

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**Факультет інформаційних технологій**

Кафедра технологій управління

Спеціальність 122 «Комп'ютерні науки»  
Освітньо-наукова програма «Управління проектами»

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА**

на тему:

«Дослідження процесів управління проектом розробки веб-сервісу для онлайн-курсів»

Студента 2-го курсу групи УП-22

Науковий керівник:

к.т.н.

Дениса АКСОЯ

Богдан ЄРЕМЕНКО

\_\_\_\_\_

(підпис студента)

\_\_\_\_\_

(дата)

\_\_\_\_\_

(підпис)

**Попередній захист:**

\_\_\_\_\_

(Висновок: «До захисту в Екзаменаційній комісії»)

Завідувач кафедри

технологій управління, к.т.н., проф. \_\_\_\_\_

Віктор МОРОЗОВ \_\_\_\_\_

(підпис)

(прізвище, ім'я)

(дата)

**Київ–2025**

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА  
Факультет інформаційних технологій**

Кафедра технологій управління

Освітній рівень Магістр

Спеціальність 122 Комп'ютерні науки

Освітньо-наукова програма Управління проєктами

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

к.т.н., проф. Віктор МОРОЗОВ

“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ року

**З А В Д А Н Н Я  
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

Студент: Аксой Денис Гюралпович

Група: УП-22

**1. Тема кваліфікаційної роботи**

«Дослідження процесів управління проєктом розробки веб-сервісу для онлайн-курсів»

Затверджена протоколом № 6 від “26” листопада 2024 року.

**2. Строк подання студенткою готової роботи** - “7” травня 2024 р.

**3. Цільова установка та вихідні дані до роботи:** дослідження сучасних підходів до управління ІТ-проєктами з фокусом на галузь EdTech, розробка концепції, технічне та організаційне моделювання, управління ресурсами, ризиками, якістю, бюджетом та етапами життєвого циклу веб-сервісу.

**4. Зміст роботи:**

аналіз ринку онлайн-освіти, визначення доцільності створення платформи, побудова концептуальної моделі продукту, опис життєвого циклу проєкту, розробка технічної архітектури, побудова ієрархічної структури робіт (WBS), опис організаційної структури, ролей та відповідальностей у команді, формалізація вимог, побудова календарного плану та діаграми Ганта, створення фінансової моделі, планування ресурсів, оцінка ризиків, управління якістю та контроль за виконанням. Проведено математичне моделювання продуктивності системи, застосовано теорію черг, здійснено аналіз витрат і доходів.

**5. Перелік графічного матеріалу (слайдів):**

титульна сторінка, постановка проблеми, мета проєкту, структура команди, дорожня карта розробки, WBS-модель, інфраструктура розгортання, діаграма Ганта, SWOT-аналіз, PEST-аналіз, інтелектуальна карта, структура бази даних, схема архітектури системи, модель черги, фінансовий графік, карта стейкхолдерів, висновки.

## 6. Календарний план виконання роботи:

№ з/п	Назва частин роботи	Виконання роботи
1	Вивчення літературних джерел з предмету дослідження	12.01.25-20.01.25
2	Збір і вивчення матеріалів досліджуваного підприємства	21.01.25-09.03.25
3	Складання розгорнутого плану кваліфікаційної роботи	10.03.25-15.03.25
4	Ознайомлення наукового керівника з планом кваліфікаційної роботи	16.03.25
5	Підготовка розділу 1	17.03.25-29.03.25
6	Підготовка розділу 2	30.03.25-10.04.25
7	Підготовка розділу 3	11.04.25-20.04.25
8	Підготовка розділу 4	20.04.25-28.04.25
9	Оформлення кваліфікаційної роботи	29.04.24-06.05.25
10	Передача кваліфікаційної роботи науковому керівникові	07.05.25
11	Передача кваліфікаційної роботи рецензенту для рецензування	09.05.25
12	Захист кваліфікаційної роботи	26.05.25-28.05.25

Дата видачі завдання “27” листопада 2025 р.

Керівник роботи: к.т.н. Богдан Єременко

---

(підпис)

Завдання прийняв до виконання:  
студент групи УП-22 Денис Аксой

---

(підпис)

## ЗМІСТ

<b>АНОТАЦІЯ.....</b>	<b>6</b>
<b>ТАБЛИЦЯ СКОРОЧЕНЬ ТА ПОЯСНЕНЬ.....</b>	<b>7</b>
<b>ВСТУП.....</b>	<b>8</b>
<b>РОЗДІЛ 1. АНАЛІЗ СУЧАСНОГО СТАНУ ТА ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ВЕБ-СЕРВІСУ ДЛЯ ОНЛАЙН-КУРСІВ.....</b>	<b>11</b>
1.1. Аналіз предметної галузі та тенденцій розвитку онлайн-освіти в Україні та світі.....	11
1.2. Характеристика ринку веб-сервісів для онлайн-курсів та його ключових гравців.....	16
1.3. Огляд сучасних наукових підходів до управління проектами в ІТ та EdTech.....	20
1.4. Визначення проблемної області та обґрунтування інноваційного потенціалу проекту.....	22
1.5. Формулювання мети, завдань та вимог до функціоналу веб-сервісу онлайн-курсів.....	25
<b>РОЗДІЛ 2. РОЗРОБКА КОНЦЕПЦІЇ ТА ІТ-АРХІТЕКТУРИ ВЕБ-СЕРВІСУ ДЛЯ ОНЛАЙН-КУРСІВ.....</b>	<b>29</b>
2.1. Концепція веб-сервісу онлайн-курсів та його цільова аудиторія.....	29
2.2. Функціональні можливості та переваги майбутнього сервісу.....	33
2.3. UX/UI-проекткування: принципи інтерфейсу та користувацького досвіду....	36
2.4. Архітектура та технологічна структура системи.....	39
2.5. Структура бази даних та інформаційна модель сервісу.....	43
2.6. Математична модель процесів взаємодії в системі.....	47
<b>РОЗДІЛ 3. УПРАВЛІННЯ СТРУКТУРОЮ, КОМАНДОЮ ТА РЕСУРСАМИ ПРОЄКТУ СТВОРЕННЯ ВЕБ-СЕРВІСУ ДЛЯ ОНЛАЙН- КУРСІВ.....</b>	<b>55</b>
3.1. Формулювання мети, концепції та стратегічного бачення проекту.....	55
3.2. Аналіз конкурентного та внутрішнього середовища.....	57
3.3. Дерево цілей проекту та стратегія досягнення результатів.....	62

3.4. Управління змістом проєкту: планування, декомпозиція, беклог.....	64
3.5. Організаційна структура та управління командою.....	67
3.6. Управління часом: етапи, спринти, контроль виконання.....	69
3.7. Фінансове планування і моделювання ефективності.....	72
<b>РОЗДІЛ 4. МОНІТОРИНГ, КОНТРОЛЬ І РЕАЛІЗАЦІЯ ПРОЄКТУ СТВОРЕННЯ ВЕБ-СЕРВІСУ ДЛЯ ОНЛАЙН-КУРСІВ.....</b>	<b>77</b>
4.1 Розгортання та впровадження веб-сервісу.....	77
4.2 Імплементация функціональності та валідація рішень.....	82
4.3 Система контролю прогресу та управління відхиленнями.....	85
4.4 Управління командною взаємодією та цифровими інструментами.....	87
4.5 Управління стейкхолдерами та стратегічна комунікація.....	90
<b>ВИСНОВКИ.....</b>	<b>94</b>
<b>ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ.....</b>	<b>96</b>
<b>ДОДАТОК А.....</b>	<b>100</b>
<b>ДОДАТОК Б.....</b>	<b>102</b>
<b>ДОДАТОК В.....</b>	<b>104</b>
<b>ДОДАТОК Г.....</b>	<b>106</b>

## АНОТАЦІЯ

кваліфікаційної роботи магістра на тему:

### «Дослідження процесів управління проектом розробки веб-сервісу для онлайн-курсів»

Студент: Аксой Денис Гюралпович.

Науковий керівник: Єременко Богдан Михайлович.

Рік захисту - 2025.

Тема даної кваліфікаційної роботи – «Дослідження процесів управління проектом розробки веб-сервісу для онлайн-курсів».

*Метою* даної роботи є розробка концепції, планування та впровадження IT-проекту з розробки освітньої платформи, яка дозволяє користувачам проходити навчальні модулі онлайн, виконувати інтерактивні завдання, проходити тести та отримувати електронні сертифікати.

*Ціллю* проекту є побудова управлінської моделі для проекту створення веб-сервісу онлайн-курсів, яка забезпечує ефективну організацію робіт на всіх етапах життєвого циклу.

*Наукова новизна роботи* полягає в комплексному поєднанні методів управління проектами, інструментів програмної інженерії, ринкової аналітики та математичного моделювання в одному дослідженні.

Об'єкт дослідження – процес управління розробкою інформаційної системи в галузі дистанційної освіти. Предмет дослідження – процеси управління вимогами, змістом, термінами, ресурсами, ризиками, якістю, зацікавленими сторонами й бюджетом в межах IT-проекту.

Кваліфікаційна робота складається з анотації, вступу, основної частини, яка включає чотири розділи, висновків, списку використаних джерел та додатків.

У роботі проаналізовано тенденції розвитку EdTech-сектору, проведено аналіз конкурентів та сформовано стратегічне бачення розвитку сервісу. Здійснено PEST-аналіз, SWOT-аналіз, побудовано дерево цілей, дорожню карту та структуру WBS. Розроблено концепцію архітектури веб-сервісу, базу даних та інформаційну модель. Проект побудовано на основі гібридного управління – поєднано підходи PMBOK, SCRUM і DevOps. Реалізовано моделі планування часу, управління командою, фінансове моделювання з прогнозом грошових потоків і точкою беззбитковості. Виконано моделювання навантаження на систему з використанням теорії черг M/M/1, а також оцінено ризики і побудовано карту стейкхолдерів. Проведено деталізоване ресурсне планування, сформовано беклог і календар виконання проекту.

Продуктом проекту є веб-платформа для масових відкритих онлайн-курсів. Сервіс включає такі функції: особисті кабінети, модуль проходження курсів, тести, форуми, аналітику прогресу, панель адміністратора, інструменти створення контенту.

Робота містить 99 сторінок без урахування додатків, 25 рисунків, 39 таблиць. Додатки складають 9 сторінок.

**Ключові слова:** EdTech, управління IT-проектом, онлайн-курси, Scrum, веб-сервіс, DevOps, MVP, база даних, гейміфікація, персоналізація, CI/CD, WBS, інклюзивний UX, модель черг, KPI, цифрова трансформація освіти.

## ТАБЛИЦЯ СКОРОЧЕНЬ ТА ПОЯСНЕНЬ

Скорочення	Пояснення
IT	Інформаційні технології
EdTech	Educational Technologies (освітні технології)
MVP	Minimum Viable Product (мінімально життєздатний продукт)
UX	User Experience (користувацький досвід)
UI	User Interface (користувацький інтерфейс)
CI/CD	Continuous Integration / Continuous Deployment (безперервна інтеграція/доставка)
PM	Проектний менеджер
WBS	Work Breakdown Structure (ієрархічна структура робіт)
DB	Database (база даних)
API	Application Programming Interface (інтерфейс прикладного програмування)
AWS	Amazon Web Services (хмарна інфраструктура)
KPI	Key Performance Indicators (ключові показники ефективності)
ROI	Return on Investment (рентабельність інвестицій)
AI	Artificial Intelligence (штучний інтелект)
QA	Quality Assurance (забезпечення якості)
GDPR	General Data Protection Regulation (Загальний регламент захисту даних ЄС)
SCRUM	Методологія гнучкої розробки ПЗ
PS	Product Stakeholders (зацікавлені сторони продукту)
DBMS	Database Management System (система управління базами даних)
KISS	Keep It Simple and Straightforward (принцип простоти в UX/UI)

## ВСТУП

Стрімкий розвиток цифрових технологій та глобальна трансформація освітніх процесів стали ключовими факторами формування нової парадигми навчання, де гнучкість, доступність та персоналізація відіграють вирішальну роль. У такому контексті особливого значення набуває концепція онлайн-курсів як складової сучасної екосистеми безперервної освіти. Індустрія EdTech, що об'єднує освіту та інформаційні технології, демонструє стабільне зростання у світовому масштабі, яке обумовлене в першу чергу пандемією COVID-19, яка спричинила безпрецедентний попит на дистанційне навчання, та й фундаментальними змінами у ставленні до освіти в цілому.

У межах цих трансформацій виникає потреба в якісному програмному забезпеченні, яке б поєднувало зручність використання, масштабованість, адаптивність до потреб різних категорій користувачів та високу надійність. Створення подібних рішень вимагає чіткого планування, управління життєвим циклом розробки, ефективної комунікації з усіма стейкхолдерами та використання сучасних методів управління проектами. Саме тому розробка веб-сервісу онлайн-курсів є складним управлінським викликом, що вимагає інтеграції теоретичних знань і практичних навичок у сфері проектного менеджменту.

Тема магістерської роботи має актуальність, що зумовлена як національними потребами українського суспільства в умовах війни, так і загальними глобальними трендами в цифровій трансформації освіти. Вітчизняна система освіти активно шукає нові форми організації навчання, що дозволяють мінімізувати географічні бар'єри, враховувати індивідуальні потреби учнів і викладачів, а також підвищити ефективність навчального процесу. Одним із найбільш перспективних напрямів є розвиток власних онлайн-платформ, орієнтованих на україномовну аудиторію, що враховують культурний контекст і специфіку вітчизняного ринку.

Об'єктом дослідження в цій роботі є процес управління IT-проектом у сфері EdTech.

Предметом дослідження є методи, підходи, інструменти та моделі, що використовуються для організації, планування, реалізації та контролю проєкту створення веб-сервісу онлайн-курсів.

Метою дослідження є формування цілісної моделі управління проєктом створення освітнього веб-сервісу, яка охоплює всі ключові аспекти життєвого циклу проєкту від ініціалізації та визначення вимог до впровадження, тестування та масштабування продукту. Досягнення цієї мети передбачає вирішення ряду завдань: аналіз ринку та формування концепції продукту, побудова архітектури системи та інформаційної моделі, розробка плану реалізації, визначення складу та ролей команди, оцінка ризиків та їх мінімізація, визначення критеріїв успішності проєкту.

У процесі дослідження аналізуються практики управління змістом, термінами, вартістю, якістю, ризиками, зацікавленими сторонами та комунікаціями. Значну увагу приділено вибору інструментарію для реалізації кожного етапу проєкту, зокрема побудові WBS, створенню діаграм Ганта, застосуванню методів критичного шляху (CPM), використанню RACI-матриць та мапи стейкхолдерів.

Особливе місце у роботі займає математичне моделювання продуктивності системи, яке дозволяє кількісно оцінити ефективність архітектури та прийняти обґрунтовані рішення щодо масштабування та проєктування серверної інфраструктури. Такий підхід дозволяє поєднати класичні інженерні методи з управлінським баченням та забезпечити баланс між якістю сервісу та ефективністю використання ресурсів.

Наукова новизна роботи полягає у комплексному поєднанні управлінських підходів, математичних моделей, аналітики ринку та UX-орієнтованого дизайну при розробці онлайн-освітньої платформи, орієнтованої на український сегмент ринку. У межах дослідження формалізовано логіку розгортання MVP, побудовано стратегію масштабування, запропоновано KPI для оцінки ефективності, а також здійснено оцінку ймовірності успіху проєкту з урахуванням поведінкових, фінансових і технічних факторів.

Практична значущість дослідження полягає в тому, що його результати можуть бути використані для розробки та впровадження аналогічних ІТ-продуктів у сфері освіти. Методика, описана у роботі, адаптується до специфіки малого та середнього бізнесу, дозволяючи швидко масштабувати рішення при збереженні якості та стабільності.

У структурі роботи виділено чотири основних розділи. У першому розглянуто теоретичні основи управління ІТ-проєктами, класифікацію освітніх платформ, основи концептуалізації онлайн-сервісів. У другому – розроблено та обґрунтовано архітектуру та функціональну модель майбутнього сервісу, створено інформаційну модель бази даних. У третьому – деталізовано управлінські процеси: постановку цілей, організацію команди, планування, контроль, фінансування. Четвертий розділ присвячено практичній реалізації: середовищу розгортання, інструментам DevOps, управлінню якістю, ризиками, комунікаціями та оцінці результатів.

Таким чином, дана магістерська робота представляє собою комплексне дослідження управління повним життєвим циклом ІТ-проєкту, орієнтованого на створення високоякісного освітнього продукту в умовах сучасного українського ринку.

## **Розділ 1. Аналіз сучасного стану та обґрунтування доцільності розробки веб-сервісу для онлайн-курсів**

### **1.1. Аналіз предметної галузі та тенденцій розвитку онлайн-освіти в Україні та світі**

У сучасному глобалізованому світі інформаційні технології стали одним із ключових рушіїв трансформації всіх сфер людської діяльності, і освіта також не виняток. Інтенсивне впровадження цифрових рішень у галузі навчання спровокувало появу нових форматів подачі знань, змінило роль викладача і студента, трансформувало процес сприйняття, оцінки та обробки освітнього контенту. Онлайн-сервіси для організації та проведення курсів стали одним із найпотужніших інструментів сучасної освіти, надаючи можливість широкому спектру користувачів отримувати якісні знання незалежно від місця перебування, графіка чи попереднього досвіду.

Особливо стрімкого поштовху цифровізація освіти набула під час пандемії COVID-19, коли освітні системи по всьому світу були змушені масово перейти до дистанційного навчання [1]. За даними ЮНЕСКО [2], країни з недостатньо розвиненою ІКТ-інфраструктурою зазнали найбільших втрат: до третини учнів у світі взагалі не мали доступу до навчання в період закриття шкіл [2]. У відповідь уряди багатьох держав переглянули стратегії розвитку освіти: за оцінками ОЕСР [3], майже половина країн світу після 2020 року ухвалили нові стратегічні документи з цифрової освіти, більшість це зробили безпосередньо під час пандемії [3]. Це заклало основу для подальшого інтенсивного впровадження EdTech-рішень у навчальний процес. Водночас в Україні, починаючи з 2022 року, воєнний стан і внутрішні переміщення населення стало новим викликом для галузі освіти та додатковим стимулом до розвитку онлайн-платформ як рішення забезпечення доступу до освіти. У цих умовах веб-сервіси для онлайн-курсів значно спростили доступ до знань та стали необхідною складовою освітньої системи.

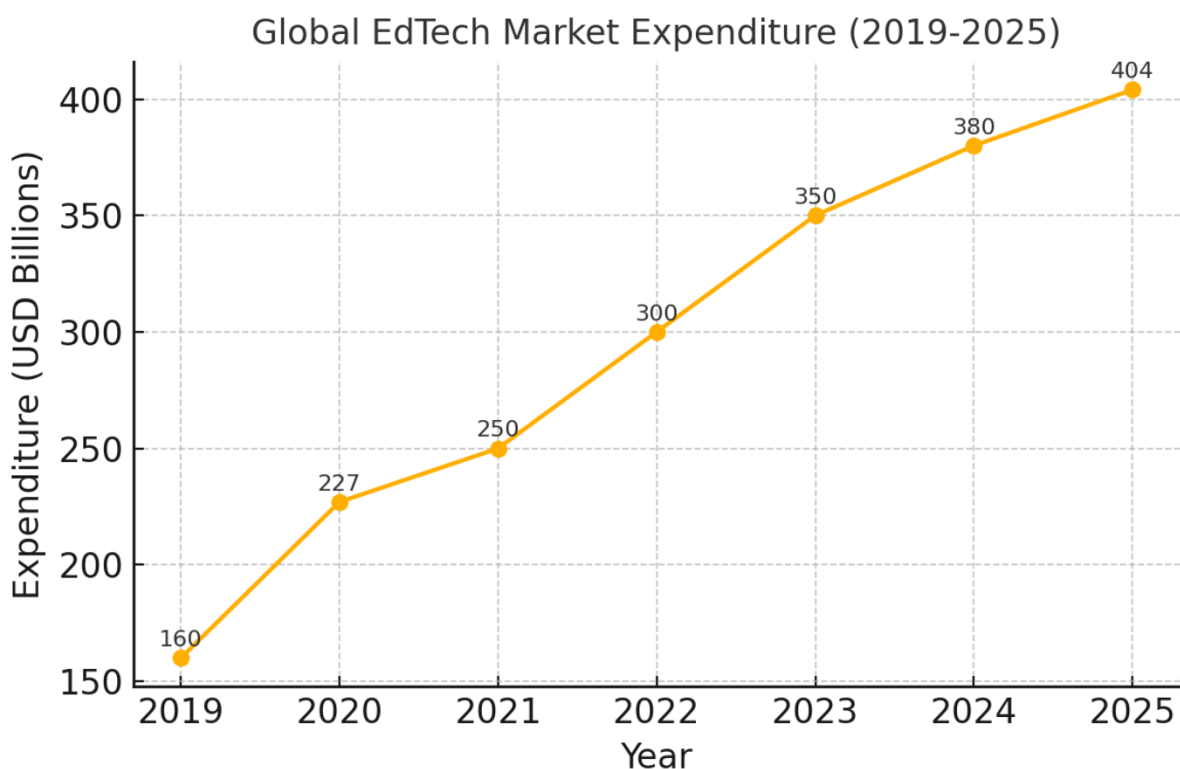


Рис. 1.1. Глобальні витрати на освітні технології у 2019-2025 рр., млрд дол. США (джерело даних: HolonIQ) [4].

Прогноз демонструє зростання ринку в 2,5 рази – з орієнтовно \$160 млрд у 2019 році до \$404 млрд у 2025 році, значною мірою за рахунок прискореної цифровізації після COVID-19. Це демонструє поступове й стабільне зростання обсягів глобального ринку освітніх технологій. Слід відзначити, що після «шокового» впровадження освітніх технологій у 2020 році можна спостерігати етап планомірної інтеграції цифрових рішень у навчальні процеси. Освітні установи поступово переходять від аварійного реагування (наприклад, екстрений перехід на Zoom) до стратегічного використання технологій, тобто вони модернізують внутрішню інфраструктуру та заранне підвищують цифрову компетентність педагогів і студентів, бо вони переосмислюють моделі навчання з урахуванням нових можливостей. Така динаміка підтверджує стратегічну важливість галузі освітніх технологій і високу інвестиційну привабливість, що надає вагомі аргументи для доцільності реалізації нових проєктів у сфері онлайн-курсів.

Серед головних світових тенденцій розвитку онлайн-освіти можна виділити наступні: широке застосування хмарних платформ і систем управління навчанням (LMS), мобільного навчання, а також активне, але в меншій мірі, впровадження технологій віртуальної/доповненої реальності (VR/AR) та штучного інтелекту (AI). Всі ці тренди свідчать про те, що технологічний фон освіти дуже швидко еволюціонує, а також пропонує нові інструменти для підвищення якості й доступності знань. У таблиці 1.1. наведено наглядну порівняльну характеристику ключових технологій та їх застосувань у сучасному навчанні. На мою думку, цифрові інструменти дозволяють зробити освіту більш гнучкою (незалежною від місця і часу) та персоналізованою.

*Таблиця 1.1.*

### **Основні технології онлайн-навчання та їх застосування**

<b>Технологія</b>	<b>Застосування у навчанні</b>	<b>Приклади та факти</b>
<b>Хмарні платформи та LMS</b>	Структура для онлайн-курсів і дистанційного навчання; забезпечують доступ до матеріалів, управління навчальним процесом, спільну роботу учасників	Використовуються в школах і ЗВО (наприклад, Moodle, Google Classroom). Хмарні сервіси дозволяють масштабувати освітні програми на тисячі слухачів одночасно
<b>Мобільне навчання</b>	Освіта за допомогою смартфонів і планшетів; доступ до курсів «на ходу», у будь-який час і місці	Різноманітні освітні мобільні додатки (Duolingo, Coursera, «Всеукраїнська школа онлайн» тощо) надають можливість навчатися поза класом. Мобільне навчання підвищує гнучкість і охоплення освітніх послуг

<b>VR/AR</b>	Іммерсивні технології для моделювання середовища навчання	VR вже застосовується у професійній підготовці (медицина, авіація тощо)
<b>Штучний інтелект (AI)</b>	Персоналізація навчання, аналіз прогресу учнів для адаптації підходів	AI-алгоритми рекомендують матеріали, адаптують навчання під студента (adaptive learning)

Онлайн-сервіси для курсів надають юзерам широкий спектр функціональних можливостей. Серед базового функціоналу є створення та адміністрування курсів, завантаження матеріалів, тестування та сертифікація. Також чимало сучасних рішень включають соціальні елементи, що підвищує залучення користувачів: чати, форуми, спільноти.

Важливою перевагою онлайн-сервісів є їхня доступність і масштабованість. Завдяки їм організатори можуть швидко оновлювати інформацію, адаптувати зміст під аудиторію, забезпечувати персоналізовану взаємодію зі студентами та швидко реагувати на зміни. Наприклад, користувачі можуть отримувати сповіщення про нові теми, зміни в розкладі, проходити автоматичні нагадування про дедлайни, що суттєво покращує організацію навчального процесу.

Рис. 1.2. ілюструє загальний рівень впровадження ключових технологій у сферу онлайн-освіти, згідно з оглядами на основі практики платформ типу Coursera, Udemy, Duolingo, Prometheus, FutureLearn.

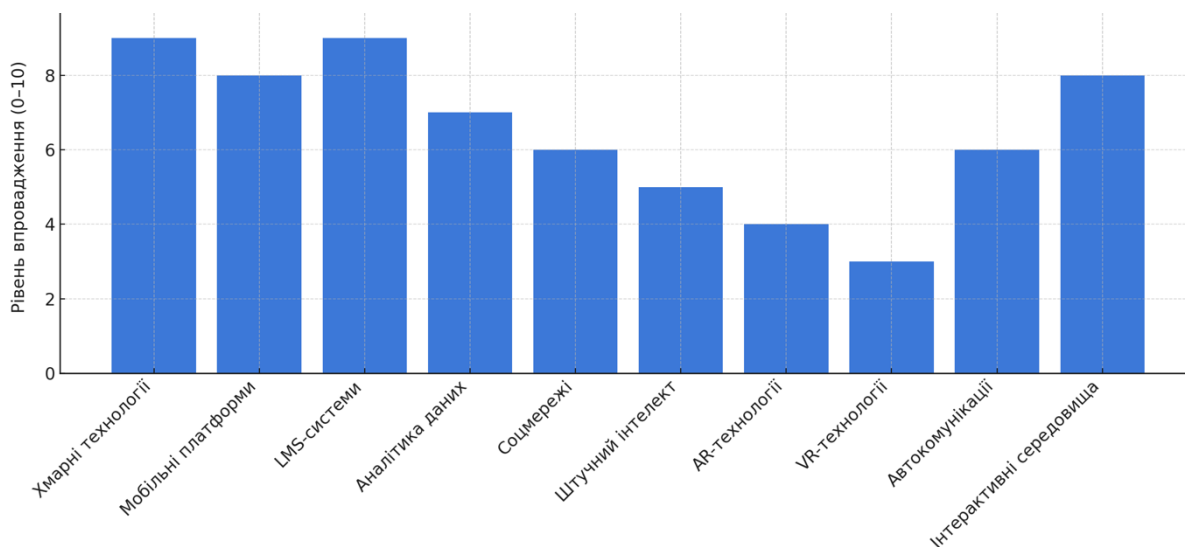


Рис. 1.2. Стан та перспективи впровадження технологій у сфері онлайн-курсів

Згідно з оцінкою, найвищий рівень впровадження спостерігається для хмарних технологій, систем управління навчанням (LMS) та мобільних платформ. Водночас використання доповненої й віртуальної реальності, штучного інтелекту (AI) і машинного навчання (ML) лише набирає обертів, хоча потенційно має величезне значення для персоналізації навчального досвіду, автоматизації оцінювання й адаптивного навчання.

Окремо слід розглянути умови і потреби розвитку онлайн-освіти в Україні у контексті останніх подій. З початком повномасштабної війни (24 лютого 2022 року) освітня система України зіткнулася з безпрецедентними викликами. Воєнний стан, загроза обстрілів та окупації територій призвели до того, що значна кількість шкіл і університетів були змушені перейти на дистанційний або змішаний формат навчання з міркувань безпеки. За оцінками ЮНІСЕФ, через COVID-19 та війну навчання українських дітей переривається вже понад п'ять років (сукупно) – це співмірно з повним курсом початкової освіти [5]. Понад 1/5 навчальних закладів країни залишаються закритими через небезпеку (відсутність бомбосховищ або пошкодження інфраструктури) [5]. Крім того, більш ніж 5 млн із 7,5 млн українських учнів і студентів були змушені змінити місце проживання через війну (внутрішнє переміщення або еміграція). У цих умовах гнучкий доступ до навчання через інтернет перетворився на життєво необхідний інструмент забезпечення права на освіту.

Таким чином, після 2020 року спостерігається унікальне поєднання факторів, що підвищують актуальність онлайн-освіти в Україні та світі: з одного боку – технологічна готовність, а з іншого – гостра суспільна потреба у гнучкому навчанні (спочатку через пандемію, а далі через війну). Глобальні тенденції синхронізуються з українськими реаліями. Це підкреслює актуальність обраного напрямку дослідження і обґрунтовує доцільність реалізації нового українського веб-сервісу для онлайн-курсів. Такий проєкт відповідає світовим трендам розвитку освіти, водночас адресуючи нагальні потреби українських учнів і студентів у безперервному та якісному навчанні в будь-яких обставинах.

## **1.2. Характеристика ринку веб-сервісів для онлайн-курсів та його ключових гравців**

Ринок освітніх веб-сервісів, орієнтованих на проведення онлайн-курсів, демонструє стрімке зростання та технологічний розвиток, що є наслідком поєднання кількох фундаментальних тенденцій – масової цифровізації в світі, а також зростання попиту на неформальну і швидку освіту. Аналіз ключових гравців цього ринку дозволить виявити спільні риси провідних платформ та унікальні підходи до організації навчального процесу, які формують конкурентні переваги кожної з них.

Серед найвідоміших міжнародних платформ для онлайн-курсів можна відзначити Coursera, Udemy, edX. У контексті українського ринку вагомому роль відіграють Prometheus – це локальний проєкт, що адаптували практики світових брендів до освітніх потреб української аудиторії.

Coursera та edX співпрацюють із університетами та іншими освітніми установами, частіше надаючи академічно орієнтований контент і можливість отримання сертифікатів [6], [7]. Слід відзначити, що Coursera пропонує тисячі курсів у партнерстві з провідними університетами, а це забезпечує високий рівень довіри та визнані сертифікати. Coursera використовує модель «freemium» – тобто базові функції доступні безкоштовно, але сертифікація або спеціалізовані програми потребують оплати. edX, заснований Массачусетським технологічним

інститутом і Гарвардом, також спеціалізується на університетських, що гарантує академічну якість контенту.

Натомість Udemу функціонує за іншим принципом. Udemу дотримується моделі відкритого маркетплейсу, що дає змогу індивідуальним експертам створювати і продавати власні курси. Це забезпечує великий спектр опцій – сотні тисяч курсів – але й певну варіативність якості, оскільки контент не проходить строгого академічного відбору.

Український ринок EdTech також представлений власними платформами, найбільшою з яких є Prometheus. Prometheus співпрацює з провідними викладачами, компаніями та міжнародними організаціями для створення онлайн-курсів, і він є локальним аналогом Coursera. Більшість курсів на платформі є безкоштовними, оскільки вони фінансуються за рахунок грантової підтримки та партнерств, і це однозначно є перевагою, адже це знижує фінансові бар'єри до навчання.

Порівняльна таблиця функціональності популярних платформ онлайн-курсів представлена нижче (таблиця 1.2.). Ця таблиця відображає ключові характеристики чотирьох платформ: Coursera, edX, Udemу та Prometheus.

*Таблиця 1.2.*

**Порівняльна характеристика функціональності платформ масових онлайн-курсів.**

<b>Платформа</b>	<b>Джерело контенту</b>	<b>Масштаб платформи</b>	<b>Модель доступу та сертифікації</b>
<b>Coursera</b>	Університети та провідні компанії-партнери	~10 000 курсів	Безкоштовний доступ до матеріалів із оплатою за сертифікат; підписка Coursera Plus
<b>edX</b>	Університети	~3 000 курсів	Безкоштовний перегляд курсу (audit); платне отримання сертифікату

<b>Udemy</b>	Незалежні експерти	~250 000 курсів; широкий тематичний діапазон	Платна разова купівля курсу (довічний доступ)
<b>Prometheus</b>	Університети та експерти з України	~100 курсів українською	Безкоштовний доступ до всіх матеріалів; сертифікація безкоштовна або за мінімальну плату

Таблиця 1.2. наглядно показує, як глобальні платформи (Coursera, edX) роблять акцент на академічній якості контенту, оскільки основним джерелом контенту є сертифіковані університети, тоді як маркетплейси на кшталт Udemy забезпечують велику кількість курсів і гнучкість у навчальних. Модель Udemy демократизує доступ до викладання, але вимагає від користувача уважної оцінки якості конкретного курсу. Український Prometheus забезпечує локальний контент і адресує потребу в навчанні рідною мовою.



Рис. 1.3. Розподіл бізнес-моделей серед EdTech-платформ

Сучасні платформи онлайн-навчання використовують різні бізнес-моделі або їх поєднання, залежно від цільової аудиторії та стратегії монетизації. Розглянемо основні типи бізнес-моделей у сфері EdTech та їх приклади (таблиця 1.3.):

Таблиця 1.3.

**Основні бізнес-моделі EdTech-платформ та приклади їх застосування**

<b>Бізнес-модель</b>	<b>Сутність</b>	<b>Приклади реалізації</b>
<b>Підписка</b>	Регулярна абонентська плата за доступ до каталогу курсів або функцій	Coursera Plus (\$59/міс); Udemy Personal Plan (\$20/міс)
<b>Маркетплейс</b>	Відкритий майданчик для сторонніх інструкторів та дохід від комісії з продажів	Udemy – 50% комісії з продажу курсів; Preply – комісія з уроків репетиторів
<b>Реклама + Premium</b>	Безкоштовний продукт із рекламою та платний тариф для вимкнення реклами	Duolingo має безкоштовні уроки з показом реклами, Duolingo Plus без реклами
<b>Інституційна (B2G/B2Edu)</b>	Продаж рішень школам, університетам	Moodle, Google Classroom; державні платформи на зразок «Всеукраїнська школа онлайн»
<b>Корпоративна (B2B)</b>	Продаж платформи або курсів компаніям для навчання співробітників; корпоративні акаунти	Coursera for Business; LinkedIn Learning Enterprise
<b>Гібридна</b>	Комбінування кількох моделей залежно від сегментів аудиторії	Більшість великих EdTech: Coursera (B2C + B2B); EdEra (B2C + B2B)
<b>Freemium</b>	Безкоштовний базовий контент; оплата за сертифікати або розширені функції	Coursera, edX

Розуміння бізнес-моделей важливе для визначення масштабованості майбутнього сервісу. Для українського стартапу в сфері EdTech доцільно розглядати гібридну модель: поєднання freemium для залучення масової аудиторії, підписки/разових платежів для монетизації. Такий підхід дозволить диверсифікувати джерела доходу і гарантувати інноваційний потенціал проєкту, маючи підхід до кількох ринкових ніш.

### **1.3. Огляд сучасних наукових підходів до управління проєктами в ІТ та EdTech**

Розробка веб-сервісу для онлайн-курсів – це складний проєкт, успіх якого значною мірою залежить від обраної методології управління. В сучасній практиці програмних проєктів домінують дві парадигми управління проєктами: традиційна каскадна модель (Waterfall) та гнучка методологія (Agile), що включає Scrum, Kanban та інші фреймворки. Ми розглянемо їхні особливості, переваги та недоліки з точки зору застосування в нашому контексті.

Підхід каскадної моделі (Waterfall) історично склався в інженерних дисциплінах і передбачає послідовне проходження фаз життєвого циклу:

Аналіз вимог → Дизайн → Реалізація → Тестування → Впровадження

В цій моделі перехід до наступної фази відбувається тільки після завершення попередньої, а повернення назад мінімізується. Серед переваг Waterfall є чітка структура, детальна початкова документація, прогнозованість термінів і бюджету в стабільних проєктах. Однак суттєвим недоліком каскадної моделі є низька гнучкість: якщо під час реалізації змінюються вимоги або з'являються нові обставини, внести корективи важко і дорого. У випадку розробки EdTech-сервісу, де вимоги можуть уточнюватися поступово (наприклад, на основі зворотного зв'язку від перших користувачів), каскадний підхід не завжди ефективний. Ризик виявлення помилок чи невідповідностей часто проявляються на стадії тестування, коли виправлення обходиться дорого і може негативно вплинути на вже виконану роботу.

На противагу цьому, для розробки веб-платформ гнучкі підходи (Agile) пропонує ряд переваг. По-перше, можна виділити гнучкість і адаптивність:

команда може швидко реагувати на зміни вимог, додавати новий функціонал або змінювати існуючий. Це критично після запуску MVP (мінімально життєздатного продукту), де будуть отримані відгуки користувачів і виникне потреба в коригуванні функцій. По-друге, завчасне виявлення проблем: регулярне тестування дозволяє знайти технічні проблеми або невідповідності очікуванням замовника на ранніх етапах. Agile передбачає тісну співпрацю із замовником на всіх етапах. Це гарантує, що розробка буде в руслі реальних потреб, а не гіпотез, закладених на початку. Scrum вважається найбільш популярним Agile-фреймворком, який передбачає спринти тривалістю 1-4 тижні, ролі Scrum-мастера та Product Owner, щоденні наради команди, беклог задач і регулярні ретроспективи. Scrum нині використовують близько 87% організацій, що впроваджують Agile, що робить його стандартом гнучкого управління проектами.

*Таблиця 1.4.*

#### **Порівняння традиційних і гнучких підходів до управління ІТ-проектами**

<b>Критерій</b>	<b>Традиційний підхід</b>	<b>Agile-підхід</b>
<b>Ключовий принцип</b>	Послідовність	Ітеративність
<b>Гнучкість змін</b>	Низька	Висока
<b>Ролі в команді</b>	Фіксовані	Гнучкі
<b>Вимоги</b>	Фіксовані на початку	Уточнюються в процесі
<b>Комунікація</b>	Формалізовано через документацію	Жива взаємодія
<b>Оцінка прогресу</b>	Звітність	Робочий продукт
<b>Управління ризиками</b>	Планується на початку, контроль зазвичай наприкінці	На кожному етапі
<b>Фокус</b>	На плані та процесі	Цінність для користувача

Для проєкту розробки EdTech-сервісу вважаю доцільним обрати Agile-підхід (Scrum). Аргументую це тим, що проєкт є інноваційний, та вимоги можуть еволюціонувати. Scrum дозволить кожні 2-4 тижні отримувати нові версії платформи і тестувати їх на користувачах. Також вважаю важливим застосувати елементи DevOps: автоматизоване тестування, безперервну інтеграцію (CI) та доставку (CD), що добре узгоджується з Agile. Це особливо корисно для проєкту веб-сервісу, де оновлення можуть відбуватися непомітно для звичайного користувача.

#### **1.4. Визначення проблемної області та обґрунтування інноваційного потенціалу проєкту**

У контексті цифрової трансформації освіти ринок онлайн-курсів переживає стрімкий розвиток, однак цей процес супроводжується низкою викликів, які залишаються не вирішеними. Саме ці структурні обмеження створюють передумови для інновацій та обґрунтовують актуальність розробки нового веб-сервісу. Відсутність глибокої локалізації, нерівномірна якість контенту, складність інтерфейсів, обмежена аналітика, а також проблеми з доступом до навчальних матеріалів без постійного з'єднання з інтернетом – це формує проблемну область.

Хоча різні освітні платформи забезпечують великий спектр курсів, вони переважно орієнтовані на глобальну аудиторію, часто залишаючи поза увагою потреби локальних користувачів. Українськомовний інтерфейс або субтитри – рідкість, що створює бар'єри для значної частини населення, особливо в умовах воєнного часу, коли освітній процес має бути доступним. Більше того, складність деяких інтерфейсів у поєднанні з перевантаженістю функціоналом ускладнює доступ до знань для аудиторій без глибокої технічної підготовки.

Аналіз потреб аудиторії показує, що ключовими напрямками інновацій, які вирізнятимуть пропонований сервіс, є персоналізація навчання, локалізація та культурна адаптація контенту, інклюзивність, безпека даних, відмінний UX/UI дизайн та гейміфікація навчального процесу. Ці потреби мають практичний

характер і пов'язані з особливостями побутових умов та й з очікуваннями цифрової зручності, притаманними більшості користувачів.

Проведений аналіз дозволяє визначити ключові напрямки інноваційного потенціалу майбутнього сервісу. Новий сервіс впровадить персоналізовані формати навчання, використовуючи штучний інтелект та машинне навчання для адаптації контенту. Інноваційність цього підходу полягає в тому, що платформа не обмежиться рекомендацією курсів, але й підлаштовуватиме курс під студента.

Наприклад, за результатами початкового тестування користувача система може пропускати відомі йому розділи або пропонувати додаткові пояснення тем, які даються важко. Провідні на ринку освітні платформи акцентують увагу на персоналізації навчання як на стратегічно важливій функції для задоволення різноманітних потреб аудиторії. Індивідуальний підхід на платформі стане вагомим конкурентним перевагою на українському ринку, де більшість конкурентів пропонують однаковий контент для всіх.

Враховання локального контексту – це критичний чинник успіху. Курси, що будуть виставлені на платформі, проходять експертизу на предмет відповідності українським реаліям. Будуть впроваджені інструменти локалізації, які дозволяють додавати субтитри, аудіодоріжки чи переклади матеріалів іншими мовами. Це відкриває можливість партнерства з міжнародними партнерами. Для українських учнів будуть використовуватися знайомі контексти (національний бізнес, історичні події, локальні дані тощо), що значно підвищить релевантність навчання. Батьки схильні більше інвестувати в платформу, яка навчає їх дітей рідною мовою, а студенти швидше зрозуміють матеріал на близьких прикладах.

Дотримання стандартів веб-доступності WCAG 2.1 з самого початку є дуже важливим аспектом для успішності платформи. Підтримка екранних дикторів для незрячих людей, субтитри для відеолекцій, продумана колірна гама для людей з порушеннями зору, зручна навігація з клавіатури тощо. Інклюзивність не обмежується фізичними особливостями, тому що вона також охоплює різні стилі навчання і соціальні групи. Платформа надасть різноманітні формати навчального матеріалу щоб кожен учень міг обрати зручний для нього

спосіб сприйняття. Інклюзивний підхід підвищить охоплення аудиторії та відповідатиме суспільному запиту на рівні можливості в освіті. Варто зазначити, що новітні технології, зокрема штучний інтелект, відкривають додаткові можливості для підвищення доступності.

З самого початку розробки будуть закладені високі стандарти кібербезпеки та захисту даних. Підхід полягає у проактивному забезпеченні відповідності міжнародним стандартам безпеки (OWASP для веб-застосунків, шифрування даних, двофакторна аутентифікація). Особлива увага – захисту персональних даних студентів: платформа відповідатиме нормам GDPR та українського законодавства про захист інформації. Зокрема, багато компаній підписують зобов'язання перед батьками і студентами про відповідальне поводження з їхніми даними. Треба гарантувати залучення до подібних ініціатив, і це однозначно стане конкурентною перевагою. Окрім захисту приватності, приділятиметься увага захисту контенту від нелегального копіювання. Безпечне середовище – це обов'язкова умова довіри користувачів, навіть якщо вона менш помітна кінцевому користувачу.

Якісний користувацький досвід (UX) – це пряма умова оптимальної функціональності, інтуїтивності, задоволеності користувача від роботи з продуктом. Сервіс впровадить user-centered design (дизайн, орієнтований на користувача), бо добре продуманий інтерфейс підвищує залученість і успішність навчання. Особлива увага приділятиметься навігації по курсу, бо студент завжди має розуміти, де він знаходиться в навчальній програмі, які вже пройшов модулі, а які попереду, мати легкий доступ до матеріалів та своїх результатів. Цей принцип лежатиме в основі розробки.

Щоб зробити навчальний процес цікавим, сервіс застосовуватиме гейміфікацію – елементи ігрового дизайну у неігровому контексті. Заплановано впровадити такі механіки як: рейтинги (за виконані завдання, активність на форумах), значки та досягнення (за завершення курсу, за відмінні результати). Важливо використовувати гейміфікацію дозовано і доречно, особливо для більш дорослої аудиторії. Ефективність гейміфікації підтверджується численними

дослідженнями: вона підвищує залученість і мотивацію учнів, а отже сприяє кращим освітнім результатам [8]. Гейміфікація в комплексі з персоналізацією (яка підлаштовує складність і темп) та хорошим UX (який не створює бар'єрів) допоможе довести студента до завершення курсу.

У перспективі, за рахунок правильної архітектури, можлива масштабована реалізація проєкту на ринки Центрально-Східної Європи. Крім того, сервіс має високий потенціал інтеграції з системами формальної освіти як допоміжний інструмент у вищих навчальних закладах або для підвищення кваліфікації працівників.

Таким чином, розробка проєкту є обґрунтованою з точки зору актуальності, ринкової потреби, технологічного прогресу та соціального запиту на доступну, зручну і ефективну онлайн-освіту.

### **1.5. Формулювання мети, завдань та вимог до функціоналу веб-сервісу онлайн-курсів**

Управління проєктом розробки веб-сервісу для онлайн-курсів вимагає чітко визначеної мети, системи взаємопов'язаних завдань, а також ретельно сформульованих вимог до функціоналу продукту. Ці аспекти є основними критеріями для ефективного проєктування, адже вони впливають на якість, зручність використання, надійність платформи. На основі визначених пріоритетів було сформульовано вимоги до функціональності майбутнього сервісу.

Метою даного дослідження є проєктування, обґрунтування функціональності та управління процесами розробки веб-сервісу для проведення онлайн-курсів, орієнтованого на українськомовну аудиторію, з урахуванням сучасних тенденцій у сфері EdTech. Ця мета передбачає створення функціонального програмного продукту й досягнення більш широкої цілі, якою є розвиток інноваційної освітньої платформи, яка відповідатиме сучасним викликам цифрової епохи.

Платформа повинна підтримувати реєстрацію та автентифікацію користувачів (студентів, викладачів, адміністраторів). Необхідно реалізувати

різні ролі з відповідними правами: студент (проходження курсів, участь в обговореннях), викладач курсу (створення контенту, управління своїми курсами, оцінювання), адміністратор (модерація контенту і користувачів). Облікові записи повинні містити профіль з основними даними та показниками прогресу.

Сервіс також має надавати зручний каталог курсів з фільтрацією за категоріями, рівнем, мовою, тощо. Для кожного курсу необхідна сторінка з описом, програмою, інформацією про викладача, відгуками. Система управління контентом має дозволяти авторам завантажувати відеолекції, тексти лекцій, презентації, завдання, тести. Платформа повинна підтримувати різні формати відео, текстові файли. Бажано мати вбудований простий інструмент для створення тестових питань.

Для студентів платформа має забезпечувати LMS-функціонал з відстеженням прогресу проходження курсу. Має бути дашборд студента, де видно поточні курси, відсоток виконання, оцінки. Для кожного курсу має існувати структура модулів, позначки про завершення кожного. Також потрібно реалізувати форум або дошку обговорень для кожного курсу, щоб студенти могли ставити питання, а викладач чи інші учасники відповідати.

Система має вміти збирати дані про прогрес студента, успіхи та складнощі, і на основі цього надавати рекомендації. Ці вимоги вимагатимуть розробки алгоритмів AI/ML, щоб забезпечити відмінність платформи за якістю навчального процесу.

Платформа інтегрує модуль гейміфікації з нарахуванням балів за активність, системою рівнів або рейтингу, системою досягнень з привабливим дизайном (наприклад, “Помічник спільноти” за відповіді іншим). Також важливо мати нагадування та автоматичні сповіщення – система повинна надсилати email або push-сповіщення про новий контент в улюблених категоріях. Ці механізми підвищать утримання користувачів.

Сервіс повинен підтримувати API для інтеграції з платіжними сервісами. Важливим є інтеграція з соціальними мережами для реєстрації (автентифікація

через Google, Facebook) та експорт досягнень (поділитися у LinkedIn чи інші соціальні мережі отриманим сертифікатом).

Хоча основна увага виділяється на функціонал, новий сервіс має відповідати деяким нефункціональним вимогам: забезпечувати одночасну роботу великої кількості користувачів, захищеність, резервне копіювання даних. Час відгуку інтерфейсу має бути мінімальний, завантаження сторінок повинно бути швидким навіть на мобільному інтернеті, система повинна витримувати пікові навантаження.

*Таблиця 1.5.*

### **Вимоги до функціоналу веб-сервісу онлайн-курсів**

<b>Категорія</b>	<b>Опис вимог</b>	<b>Приклади реалізації</b>
<b>Реєстрація та ролі</b>	Підтримка студентів, викладачів, адміністраторів; окремі права доступу.	RBAC, Firebase Auth
<b>Управління контентом</b>	Завантаження відео, текстів, програм курсів.	CMS, відеохостинг, текстові редактори
<b>LMS-функціонал</b>	Прогрес, оцінки, завершення модулів, структура курсів.	Moodle-подібна логіка, індикатори прогресу
<b>Спільнота</b>	Форуми, коментарі під модулями, відповіді викладачів.	Дошка обговорень
<b>Рекомендації та аналітика</b>	AI-аналіз прогресу, збір статистики.	Google Analytics
<b>Гейміфікація</b>	Бали, бейджі, рівні, статуси за активність.	Рейтингова система

<b>Нотифікації</b>	Email/push-повідомлення, нагадування про контент.	Firestore Cloud Messaging, SendGrid
<b>API / інтеграції</b>	Платежі, соцмережі, експорт сертифікатів.	Stripe API, LinkedIn API
<b>Нефункціональні вимоги</b>	Надійність, безпека, масштабованість, швидкий відгук.	HTTPS, резервне копіювання, AWS

Перераховані вимоги утворюють основу технічного завдання на розробку платформи. Структура вимог відображає як функціональні можливості, так і характеристики якості.

Відповідно до сформульованої мети дослідження, для реалізації проєкту створення веб-сервісу для онлайн-курсів необхідно вирішити наступні завдання:

1. Провести аналіз предметної галузі та дослідити стан ринку онлайн-освіти в Україні та світі; здійснити характеристику основних типів EdTech-платформ; обґрунтувати доцільність створення веб-сервісу на основі виявлених проблем та ринкових можливостей;
2. Сформулювати концепцію майбутнього продукту, визначити цільову аудиторію та ключові функціональні можливості;
3. Розробити технічну архітектуру платформи та структуру бази даних;
4. Побудувати математичну модель процесів взаємодії в системі;
5. Визначити організаційну структуру команди проєкту та модель взаємодії;
6. Розробити WBS, календарний план, план спринтів та провести аналіз ризиків;
7. Реалізувати MVP-продукт у хмарному середовищі з використанням сучасних інструментів DevOps;
8. Здійснити фінансове планування та оцінку ефективності з урахуванням ROI, точки беззбитковості та індексу PSI.

## Розділ 2. Розробка концепції та іт-архітектури веб-сервісу для онлайн-курсів

### 2.1. Концепція веб-сервісу онлайн-курсів та його цільова аудиторія

Розвиток ринку EdTech і глобальна цифровізація освіти спровокували зростання популярності онлайн-курсів. Значний стрибок у використанні таких сервісів стався після пандемії COVID-19, яка прискорила масовий перехід до дистанційного навчання. Ці тенденції мають глибші витoki – від поширення ідеї навчання впродовж життя до зростання цифрової грамотності населення. Водночас, попит на якісний онлайн-контент, особливо українською мовою, залишається недостатньо задоволеним. Метою цього підрозділу є формулювання концепції нового веб-сервісу онлайн-курсів, обґрунтування його доцільності та окреслення цільової аудиторії.

Глобальний ринок EdTech оцінюється у сотні мільярдів доларів США. Сегмент масових відкритих онлайн-курсів у 2030 році, за прогнозами, досягне понад 411,6 млрд дол. США, а кількість користувачів перевищила 220 млн уже у 2021 році. Пандемія сприяла залученню понад 40 млн нових користувачів лише за 2020 рік. Ці цифри підтверджують високий потенціал для зростання [9].

В Україні вже працюють відомі гравці – Prometheus, EdEra, «Всеукраїнська школа онлайн», але вони здебільшого зосереджені на безкоштовному доступі та соціальній місії, водночас поступаючись закордонним аналогам у зручності, AI-функціоналі, персоналізації. Зарубіжні платформи часто недоступні через мовний бар'єр або високу вартість преміум-функцій.

*Таблиця 2.1.*

#### Ринкові тренди EdTech-платформ

Назва платформи	Цінова категорія	Модель монетизації	AI-функції	Українська локалізація
Coursera	Висока	Підписка + сертифікат	Так	Ні
Udemy	Середня	Оплата за курс	Обмежено	Ні

Продовження табл. 2.1.

<b>edX</b>	Середня	Freemium	Так	Ні
<b>Prometheus</b>	Безкоштовна	Державне фінансування	Відсутні	Так
<b>EdEra</b>	Безкоштовна	Гранти	Обмежено	Так

Виходячи з цієї аналітики, доречно обрати стратегію нового веб-сервісу фріміум-модель із акцентом на україномовну підтримку, простий інтерфейс, адаптивність до мобільних пристроїв та AI-персоналізацію навчального шляху. Сервіс розраховано на широке охоплення, з поступовим впровадженням платних функцій (сертифікація, розширена аналітика, додаткові модулі).

Цільова аудиторія веб-сервісу поділяється на кілька груп:

1. **Студенти** – для підтримки навчання або здобуття додаткових навичок;
2. **Дорослі користувачі** – перекваліфікація, розвиток кар'єри;
3. **Викладачі та контент-креатори** – створення і монетизація власних курсів;
4. **Організації та бізнес** – внутрішні тренінги й професійне навчання.

Таблиця 2.2.

### Ролі користувачів веб-сервісу онлайн-курсів

Роль користувача	Функціонал/Повноваження
<b>Гість</b>	Перегляд опису курсів, реєстрація
<b>Студент</b>	Запис на курси, проходження матеріалів, виконання тестів, перегляд результатів
<b>Викладач</b>	Створення курсів, завантаження матеріалів, оцінювання студентів, управління контентом
<b>Адміністратор</b>	Модерація контенту, управління користувачами, аналітика, налаштування системи

Поділ на ролі студента, викладача, адміністратора і гостя дозволяє адаптувати інтерфейс і функціонал до потреб кожної категорії.

Щоб обґрунтувати доречність впровадження платформи в умовах українського ринку, проведено PEST-аналіз (таблиця 2.3.):

Таблиця 2.3.

### PEST-аналіз проєкту

Фактор	Опис впливу
Політичні	Державна політика цифровізації освіти; підтримка онлайн-навчання під час війни
Економічні	Нестабільна економічна ситуація; обмежені фінансові можливості користувачів
Соціальні	Поширення моделі навчання впродовж життя; відкритість до онлайн-освіти
Технологічні	Розвиток AI, мобільних технологій, швидкісний інтернет; популяризація EdTech

Політична підтримка цифровізації, соціальна відкритість до онлайн-освіти та наявність технічних передумов формують сприятливе середовище для запуску подібного продукту. Основним ризиком залишається економічна нестабільність, що враховується через адаптивну цінову політику.

На етапі планування сервісу важливо враховувати типові бар'єри, з якими стикаються користувачі. До найбільш поширених проблем належать: відсутність локалізації, низька адаптація до мобільних пристроїв, складний інтерфейс, недостатня персоналізація.

Метод Парето (рис. 2.1.) дозволяє виділити найкритичніші з них:

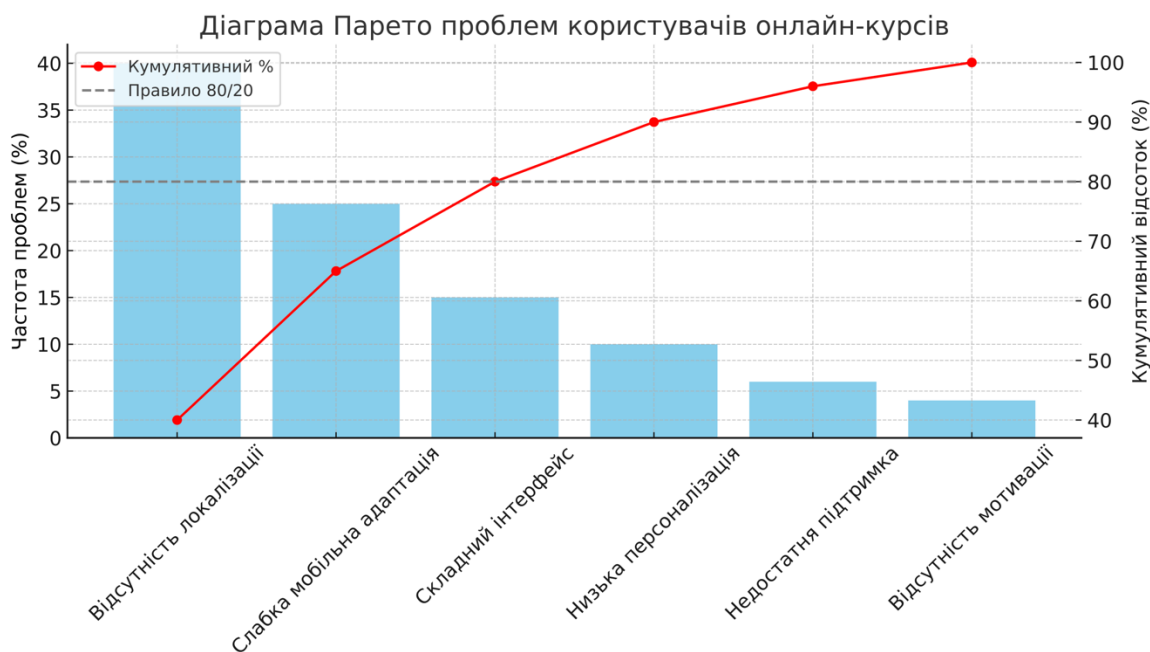


Рис. 2.1. Діаграма Парето проблем користувачів онлайн-курсів

На основі вищезгаданого аналізу запропоновано таку концепцію веб-сервісу:

1. **Доступність** – інтерфейс простий, підтримує різні пристрої.
2. **Інклюзивність** – повна україномовна локалізація, підтримка користувачів із базовою цифровою грамотністю.
3. **Персоналізація** – AI-рекомендації, адаптація навчального маршруту до дій користувача.
4. **Інтерактивність** – система досягнень, тести, дискусійні форуми.
5. **Масштабованість** – гнучка архітектура, що дозволяє інтеграцію нових партнерів, курсів, модулів.

Окрім комерційної мети, веб-сервіс виконує й соціальну функцію. Під час пандемії понад 1,6 млрд учнів у світі були позбавлені звичного навчання [10]. Платформи онлайн-курсів довели свою роль як інструменту для забезпечення безперервності освіти. Проєкт сприятиме зменшенню цифрової нерівності в Україні, поширенню освіти в регіонах, де традиційна система є менш доступною.

## 2.2. Функціональні можливості та переваги майбутнього сервісу

Проектування функціоналу веб-сервісу онлайн-курсів потребує не лише технічної деталізації, але й стратегічного обґрунтування. У цьому контексті критично важливим є врахування потреб трьох основних груп стейкхолдерів: кінцевих користувачів, авторів контенту та адміністраторів/власників платформи. Функціональні можливості сервісу повинні відповідати очікуванням цільової аудиторії, та також і забезпечити масштабованість, гнучкість та адаптивність до змін середовища.

Ключовим принципом розробки майбутнього веб-сервісу є клієнтоорієнтований дизайн (user-centered design). Цей підхід передбачає побудову логіки системи навколо реальних сценаріїв використання та поведінкових моделей аудиторії.

*Таблиця 2.4.*

### Вимоги користувачів до функціоналу веб-сервісу

Категорія користувачів	Ключові вимоги
Студенти	Мобільність, доступність, персоналізовані рекомендації, простота інтерфейсу, аналітика прогресу, гейміфікація
Викладачі	Інструменти для створення курсів, керування групами, тестування, аналітика студентів, взаємодія через форуми
Адміністрація/менеджери	Моніторинг активності, формування звітів, контроль контенту, управління доступом, інтеграція з іншими системами

Веб-сервіс структурується у вигляді кількох ключових функціональних блоків, кожен з яких орієнтований на конкретну групу завдань і тип користувача.

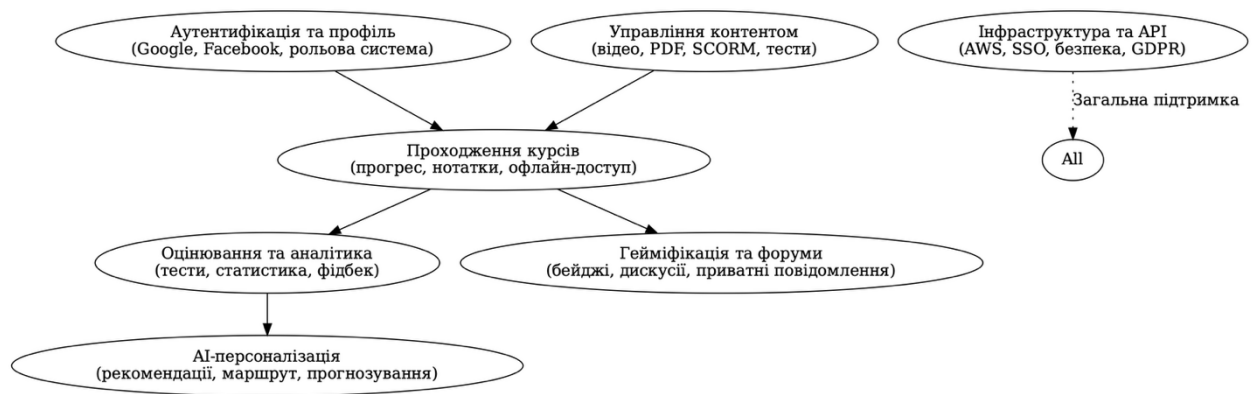


Рис. 2.2. Діаграма структури функціональних блоків системи

Аутентифікація та управління профілем реалізується через підтримку багаторольової автентифікації (студент, викладач, адміністратор, гість), а також інтеграцію з обліковими записами Google і Facebook. Користувачам надається персоналізована панель, де вони можуть бачити свій прогрес, доступні курси, повідомлення та налаштування, здійснювати самостійне керування профілем, змінювати аватар, мову інтерфейсу, а також відновлювати пароль.

Управління навчальним контентом передбачає можливість завантаження відео, PDF-документів, SCORM та HTML-контенту. Структура курсів підтримує поділ на модулі, теми, тести й домашні завдання, а також колаборативну розробку курсів викладачами з студентами. Доступ до курсів можна обмежити – вони можуть бути відкритими, приватними, з паролем або за запрошенням.

Інтерфейс студента забезпечує зручне проходження курсів, включаючи візуалізацію прогресу у формі шкали, доступ до матеріалів за темами та офлайн-режим (з можливістю попереднього завантаження). Користувачі можуть залишати нотатки до модулів і отримувати автоматичні нагадування. По завершенні курсу учасникам надаються цифрові сертифікати.

Система тестування й оцінювання дозволяє створювати тести з автоматичним або ручним оцінюванням. Викладачі мають доступ до панелі прогресу, що включає результати, рівень активності студентів та виявлення слабких тем. Доступна загальна статистика по курсу та звітність для адміністрації, а також динамічна система зворотного зв'язку – коментарі й опитування.

Інтелектуальні алгоритми персоналізації використовують поведінковий аналіз для рекомендації курсів, виявляють студентів, що відстають, та пропонують адаптивні маршрути навчання. Додатково можлива інтеграція AI-модуля для прогнозування ймовірності завершення курсу кожним студентом.

Соціальні інструменти та гейміфікація сприяють залученості, адже передбачено форуми для обговорень, приватні повідомлення, а також систему бейджів, досягнень і рівнів, що мотивує до активності.

Технічна інфраструктура передбачає API-інтеграції для підключення платіжних систем, підтримку SSO та корпоративних LMS. Хостинг здійснюється в хмарному середовищі (наприклад, AWS) з використанням CDN і безпечного зберігання даних. Система відповідає міжнародним стандартам безпеки та забезпечує відповідність вимогам GDPR.

Пропонований веб-сервіс має низку конкурентних переваг серед яких є: повна локалізація, фріміум-модель монетизації, інклюзивність, розширена аналітика і гейміфікація, проста створюваність курсів.

Функціональні переваги для трьох груп стейкхолдерів представлені в таблиці 2.5.:

*Таблиця 2.5.*

**Вигоди для основних типів користувачів платформи**

<b>Стейкхолдери</b>	<b>Переваги</b>
Користувачі	Гнучкість графіку, офлайн-режим, рекомендації, мотиваційна система
Викладачі	Простий інтерфейс створення курсів, аналітика успішності, гнучке керування контингентом
Адміністратори	Повна статистика, монетизація, брендвана версія, підключення корпоративних клієнтів

Отже, запропонована концепція веб-сервісу онлайн-курсів враховує потреби цільової аудиторії, відповідає вимогам сучасних освітніх стандартів і забезпечує потенціал для масштабування в умовах високої конкуренції на ринку цифрової освіти.

### 2.3. UX/UI-проектування: принципи інтерфейсу та користувацького досвіду

UX/UI-дизайн веб-сервісу онлайн-курсів є ще одним важливим аспектом для успішності проекту, оскільки він напряду впливає на залучення, утримання та задоволеність користувачів. Інтерфейс – це не лише візуальне оформлення, а й логіка поведінки системи у відповідь на дії користувача. Саме через нього формується перше враження про платформу, визначається легкість навігації та комфорт навчання.

UX/UI-дизайн реалізовано відповідно до загально визнаних принципів, які забезпечують інтуїтивність, простоту та функціональність:

*Таблиця 2.6.*

#### UX-принципи, реалізовані у веб-сервісі

UX-принцип	Опис реалізації
Мінімалізм	Максимальна простота: мінімум елементів на екрані, однорівнева навігація
Прогнозованість та консистентність	Усі елементи мають однакову поведінку по всьому сервісу
Принцип progressive disclosure	Складні функції відкриваються поступово в міру потреби користувача
Адаптивність	Інтерфейс адаптується під екрани будь-якого розміру, зручний на смартфонах
Інтуїтивна навігація	Користувач за 1-2 кліки досягає потрібної дії (запис на курс, перегляд прогресу тощо)
Оптимізація продуктивності	Використано техніки оптимізації: завантаження відео, кешування ресурсів
WCAG-стандарти	Висока контрастність, підтримка screen reader, масштабування шрифтів
Зворотній зв'язок	Користувач отримує повідомлення про успішні дії або помилки в зрозумілій формі

User-centered design з тестуванням	Ітеративний підхід: створено прототипи, протестовано на фокус-групі, адаптовано за фідбеком
------------------------------------	---

Ключовими стали принципи мінімалізму, передбачуваності дій, візуального зворотного зв'язку та доступності. Принцип KISS (Keep It Simple and Straightforward) застосовано до побудови кожного елемента інтерфейсу. Дотримання консистентності забезпечує однакову поведінку кнопок, іконок і навігаційних патернів на всіх етапах взаємодії.

Інтерфейс проектується з урахуванням адаптивності: він автоматично підлаштовується під різні розміри екранів. Сучасні користувачі, особливо вікової категорії 16-24 роки, проводять онлайн до 70% часу з мобільних пристроїв, що робить мобільність ключовою вимогою.

Дизайн інтерфейсу реалізовано відповідно до концепції клієнтоорієнтованого дизайну (user-centered design) – орієнтиру на потреби, поведінку та досвід користувачів. Для перевірки зручності навігації створено інтерактивні прототипи основних екранів.

Застосовано концепт progressive disclosure – поступового розкриття функціоналу залежно від дій користувача. На головному екрані доступні лише основні функції: каталог курсів, профіль, мої курси. Це дозволяє знизити когнітивне навантаження та спростити перше знайомство з платформою. Детальніший функціонал наочно показаний на рисунку 2.3, де описана схема логіки користувацької навігації у веб-сервісі (User Flow).

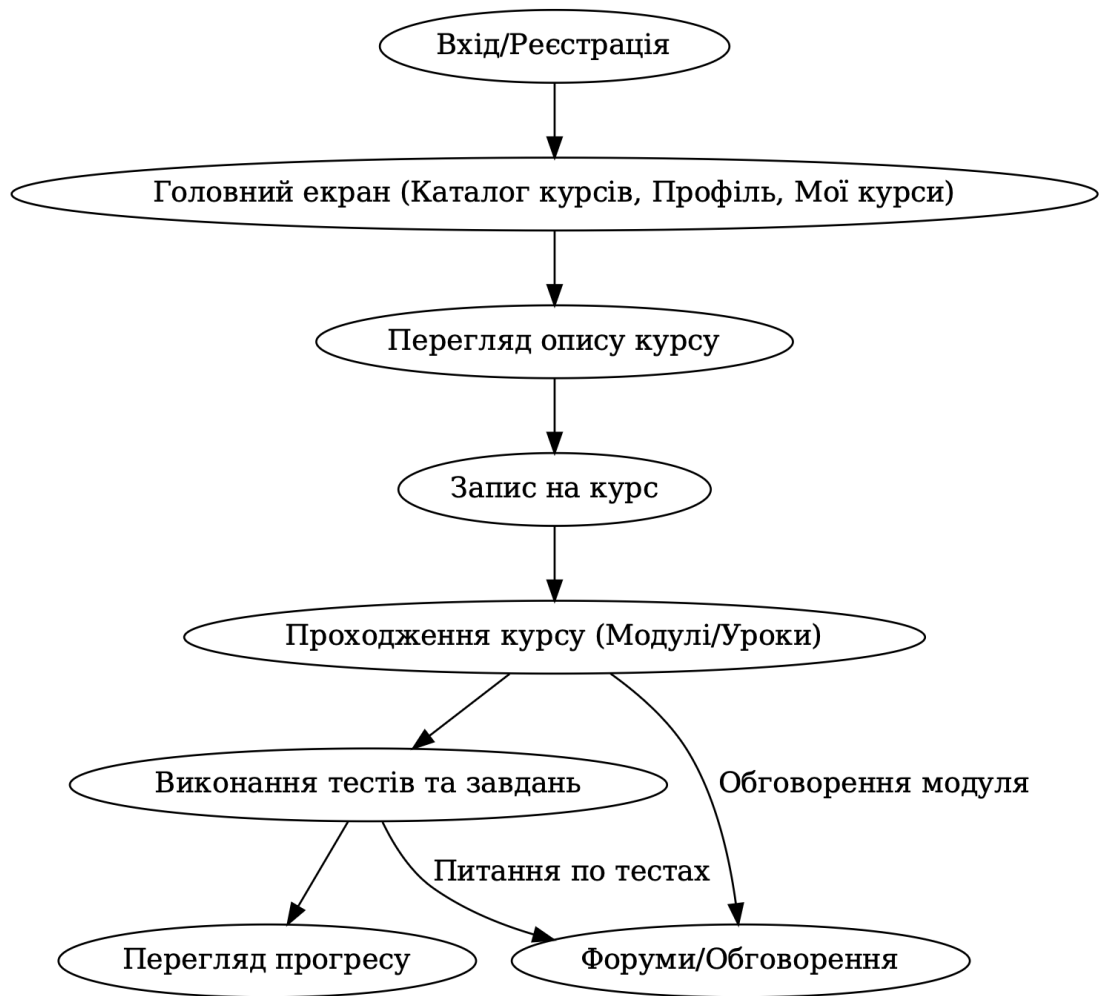


Рис. 2.3. Схема логіки користувацької навігації у веб-сервісі (User Flow)

Проектування user flow забезпечує мінімальну кількість кроків для досягнення цілі. Навігація побудована так, щоб користувач завжди знав, де він перебуває і як перейти до наступного або попереднього етапу. Основний сценарій:

Вхід → Вибір курсу → Проходження модуля → Виконання завдань →  
Перегляд результатів

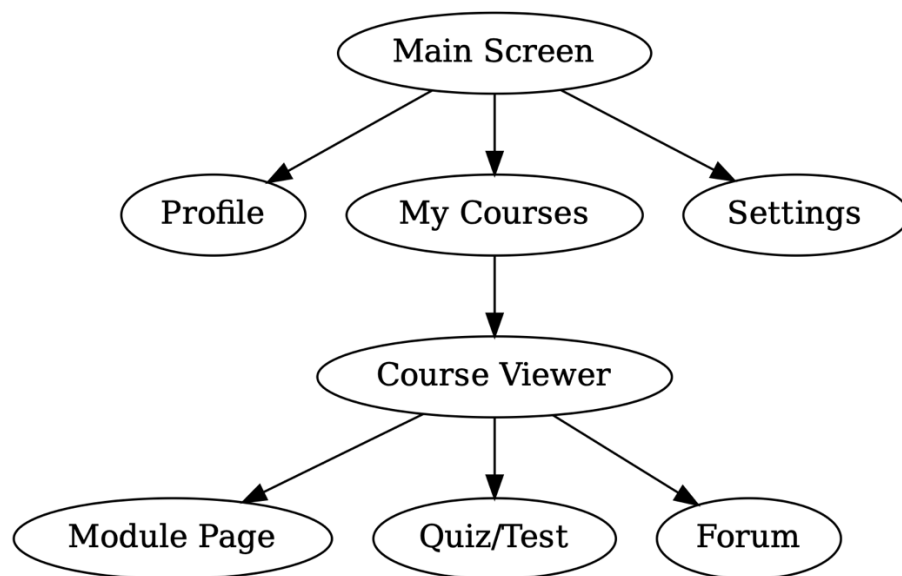


Рис. 2.4. Карта основних екранів веб-сервісу (інтерфейсна структура)

Екранна карта платформи демонструє ієрархію навігації: від головного екрана до окремих модулів і тестів. Основні екрани: Main screen, Profile, My Courses, Course Viewer, Module Page, Quiz/Test, Forum, Settings. Кожен екран має чітко визначену функцію, що забезпечує передбачуваність дій користувача.

Час відгуку інтерфейсу, стабільність роботи та швидкість завантаження являються критичними елементами UX. У реалізації використовуються кешування сторінок та автоматичне збереження відповідей користувача у випадку втрати з'єднання. Це забезпечує плавний досвід, підвищує довіру до системи та знижує ризики відтоку користувачів. Поганий UX збільшує ймовірність відмови користувача від навчання, тому інтуїтивний інтерфейс підвищує частоту завершення курсів. UX прямо впливає на показники задоволеності, що визначають лояльність аудиторії.

#### **2.4. Архітектура та технологічна структура системи**

Архітектура веб-сервісу для онлайн-курсів є ключовим елементом, що забезпечує стабільність, масштабованість, безпечність і підтримку швидкого розвитку системи. У межах цього підрозділу розглянуто концептуальну модель архітектури, логіку взаємодії модулів, вибір технологій, а також аспекти інформаційної безпеки та організації середовища розгортання.

Система реалізується як багаторівнева з чітким поділом функцій за чотирма рівнями:

1. Клієнтський рівень (Frontend) – відображення інтерфейсу, взаємодія з користувачем.
2. Серверна логіка (Backend) – обробка запитів, бізнес-логіка, персоналізація.
3. Інфраструктура – база даних, сховища, аналітика.
4. Інтеграції – авторизація, платіжні API, e-mail сервіси, аналітика.

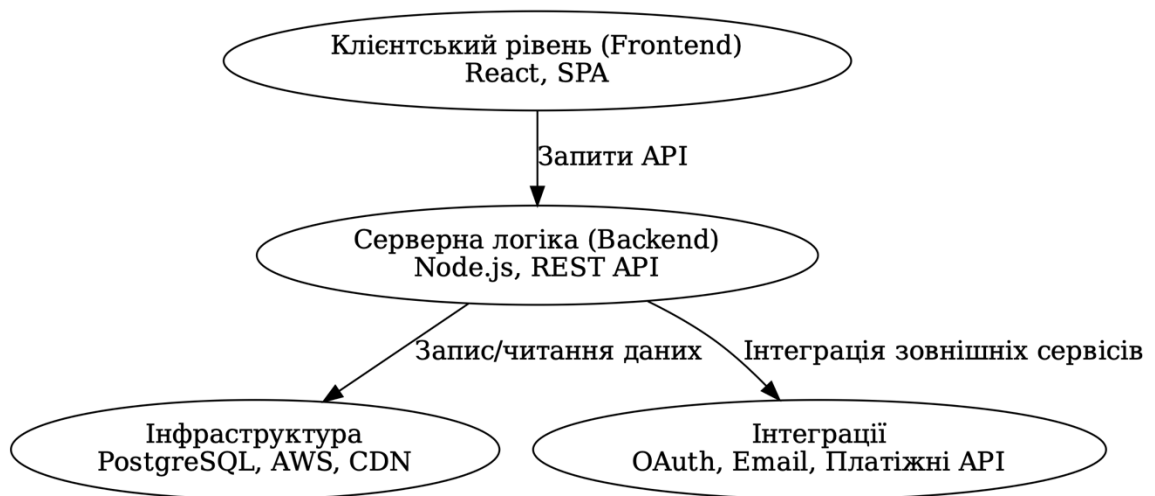


Рис. 2.5. Концептуальна модель архітектури веб-сервісу онлайн-курсів

Модель на рисунку 2.5. демонструє, як функціональні компоненти взаємодіють між собою та із зовнішніми системами. Вона є ключовою точкою референсу для подальшого технічного планування.

Комунікація між компонентами реалізується через RESTful API, що дозволяє гнучко масштабувати сервіс, інтегрувати зовнішні продукти та реалізувати мобільні додатки в майбутньому.



Рис. 2.6. Діаграма взаємодії модулів веб-сервісу

На діаграмі видно, як UI ініціює запити до API, яке у свою чергу звертається до бази даних, AI-модуля, зовнішніх API.

Для реалізації системи обрано сучасні технології, які представлені в таблиці 2.7.:

Таблиця 2.7.

### Технології для реалізації системи

Компонент	Технологія	Функція
Frontend	React.js	SPA-інтерфейс, адаптивність
Backend	Node.js	REST API, бізнес-логіка
AI-модуль	Python	ML-персоналізація, аналітика
База даних	PostgreSQL	Зберігання користувачів, курсів
Сховище	AWS	Файли, відео, PDF
Проксі-сервер	Nginx	Балансування, кешування
Авторизація	OAuth 2.0	Безпечний вхід
Email	SendGrid	Сповіщення, результати

Також у системі впроваджено комплекс заходів із забезпечення безпеки, такі рішення відповідають стандартам PMI у сфері управління ризиками:

1. Використання HTTPS для всіх з'єднань;
2. JWT-токени для авторизації;
3. Ізоляція бази даних у приватній підмережі;
4. Регулярне резервне копіювання даних;
5. Кешування популярних запитів;
6. CDN для відео й статичних ресурсів.

Застосовано двоконтурну архітектуру середовищ:

1. Production – основне середовище з реальними користувачами.

2. MVP/Test – ізольоване середовище для тестування нових функцій, що запобігає впливу на продакшн.

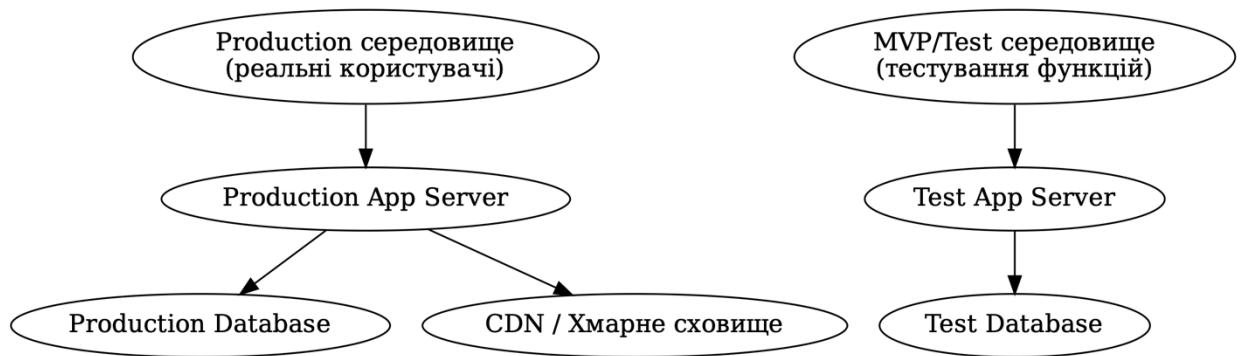


Рис. 2.7. Схема розгортання середовища (production vs test)

Така структура дозволяє апробацію змін, верифікацію нових функцій і гнучке масштабування, що відповідає підходам Lean Startup.

Поточна архітектура реалізована у вигляді модульного моноліту, проте подальший перехід до мікросервісної архітектури:

- Розподіл функціоналу на автономні сервіси;
- Можливість незалежного масштабування кожного модуля;
- Контейнеризація (Docker) для гнучкого деплою;
- API Manager для керування взаємодією між мікросервісами.

Обрана архітектура узгоджується з принципами РМВОК (PMI):

1. Scope Management – чітке розмежування компонентів дозволяє контролювати обсяг;
2. Risk Management – ізоляція модулів та тест-середовище мінімізують технічні ризики;
3. Quality Management – архітектура дозволяє контролювати якість кожного рівня;
4. Integration Management – централізоване API полегшує інтеграцію з іншими системами.

Архітектура також підтримує ідеологію agile: можливість ітеративного розширення, поступове впровадження нових модулів, робота в умовах зміни пріоритетів.

## 2.5. Структура бази даних та інформаційна модель сервісу

База даних є критичною складовою веб-сервісу онлайн-курсів, оскільки саме вона відповідає за зберігання, структурування і швидкий доступ до основних об'єктів системи: користувачів, курсів, навчальних модулів, результатів тестування, прогресу навчання, сертифікатів, обговорень та аналітики. Її грамотне проєктування визначає безпечність, стабільність і функціональність платформи.

Архітектура даних платформи базується на реляційній моделі з підтримкою зовнішніх ключів (foreign keys) та каскадними зв'язками. Такий підхід є традиційним для LMS та забезпечує:

1. Логічність і зрозумілість структури;
2. Відсутність дублювання інформації;
3. Легке розширення функціоналу;
4. Ефективне виконання агрегатних запитів;

Концептуальна модель буде побудована за участі розробників і менеджера проєкту. Буде сформовано повний словник даних, визначено обов'язкові поля, типи, зв'язки між сутностями, обмеження та валідаційні правила. Ця документація є частиною управління проєктом і служить основою для перевірки відповідності бази даних вимогам.

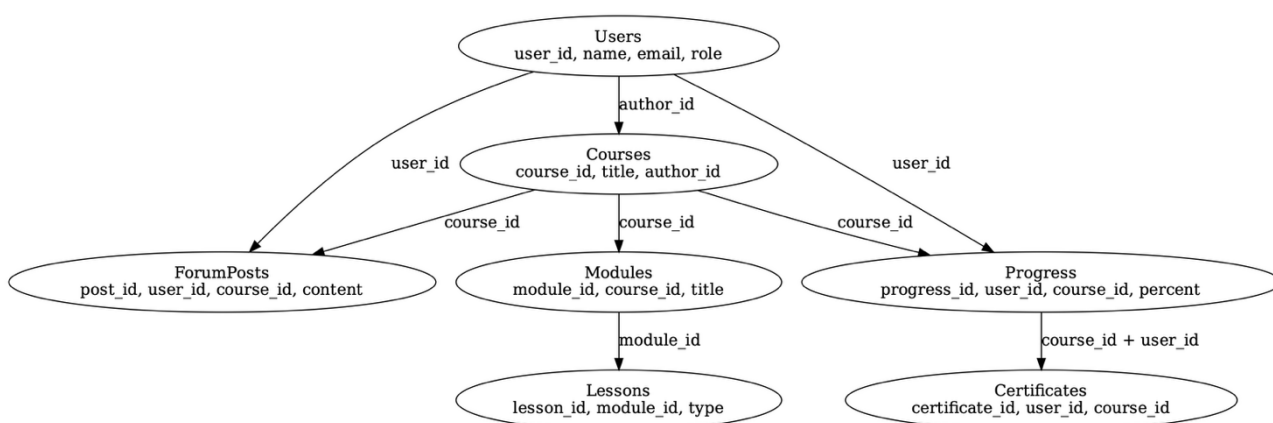


Рис. 2.8. Концептуальна модель бази даних веб-сервісу онлайн-курсів

У системі виокремлено такі основні сутності:

1. Users: ID, ім'я, email, роль (студент, викладач, адміністратор), пароль (захешований), статус активності.

2. Courses: ID, назва, опис, автор (посилання на Users).
3. Modules: ID, назва модуля, зв'язок з Courses.
4. Lessons: ID, тип (текст, відео, тест), посилання на Module, контент.
5. Progress: ID, user\_id, course\_id, percent\_completed, дата останньої активності.
6. Certificates: ID, user\_id, course\_id, дата видачі, PDF URL.
7. ForumPosts: ID, текст повідомлення, user\_id, course\_id, час публікації.

Усі зв'язки реалізовані відповідно до принципів нормалізації. Наприклад, курс має багато модулів, кожен модуль – багато уроків, один користувач може проходити багато курсів (через зв'язок з таблицею Progress). Додатково реалізовані сутності Tests, Questions, Answers, TestResults – для підтримки повного циклу оцінювання.

*Таблиця 2.8.*

### Логічна структура бази даних веб-сервісу

Сутність	Ключові поля	Зв'язки
Users	user_id, name, email, role	Один до багатьох з Courses, Progress, ForumPosts
Courses	course_id, title, description, author_id	Один до багатьох з Modules, Progress, Certificates
Modules	module_id, course_id, title	Один до багатьох з Lessons
Lessons	lesson_id, module_id, type, content_url	Має ForeignKey на Modules
Progress	progress_id, user_id, course_id, percent	ForeignKey на Users і Courses
Certificates	certificate_id, user_id, course_id, issue_date	Генерується після завершення курсу
ForumPosts	post_id, user_id, course_id, content	ForeignKey на Users і Courses

У таблиці наведено основні поля ключових сутностей, типи, первинні/зовнішні ключі, правила каскадного оновлення/видалення. Це дозволяє уникати неконсистентності та гарантує цілісність. Особливу увагу приділено таблиці Progress, яка виступає центром персоналізованого аналізу та зв'язку з AI-модулем. Вона зберігає статус проходження, дату останньої активності, темп навчання.



Рис. 2.9. Потік даних у веб-сервісі онлайн-курсів

Типовий сценарій виглядає так:

1. Користувач авторизується через OAuth 2.0 → запис у сесію;
2. Обирає курс → запит до Courses → переходить до Modules/Lessons;
3. Проходить урок → запис у Progress;
4. AI-модуль отримує оновлення → генерує рекомендації;
5. Після 100% прогресу + позитивного тесту → запис у Certificates;
6. Система надсилає PDF через SendGrid.

Модель сумісна з:

- xAPI — підтримка зовнішніх аналітичних систем, експорт результатів;
- WCAG — зберігання текстових альтернатив для зображень (для screen reader);
- LTI — потенційна інтеграція з іншими LMS.

Це забезпечує платформі потенціал для інтеграції з міжнародними освітніми екосистемами.

База даних виступає як центральне сховище системи в моделі SOA:

- Frontend (React) → REST API → Backend (Node.js/Express);
- ORM на сервері трансліює об'єкти в SQL-запити;
- Запити до PostgreSQL обробляються асинхронно;
- Аналітичні дані передаються до Google Analytics і AI-модуля (Python).

Для забезпечення цілісності збереження та обробки даних платформи, побудовано логічну архітектуру бази даних, яка відображає ключові сутності системи та зв'язки між ними. Архітектура включає таблиці користувачів, курсів, модулів, прогресу проходження, сертифікатів, а також допоміжні сутності для забезпечення авторизації, комунікації та аналітики. Взаємозв'язки реалізовані через зовнішні ключі, що забезпечує референційну цілісність.

Важливою особливістю архітектури є можливість горизонтального масштабування, що дозволяє адаптувати систему до зростання навантаження. Всі сутності логічно поділені на функціональні домени: керування курсами, взаємодія користувачів, зберігання результатів навчання, генерація сертифікатів.

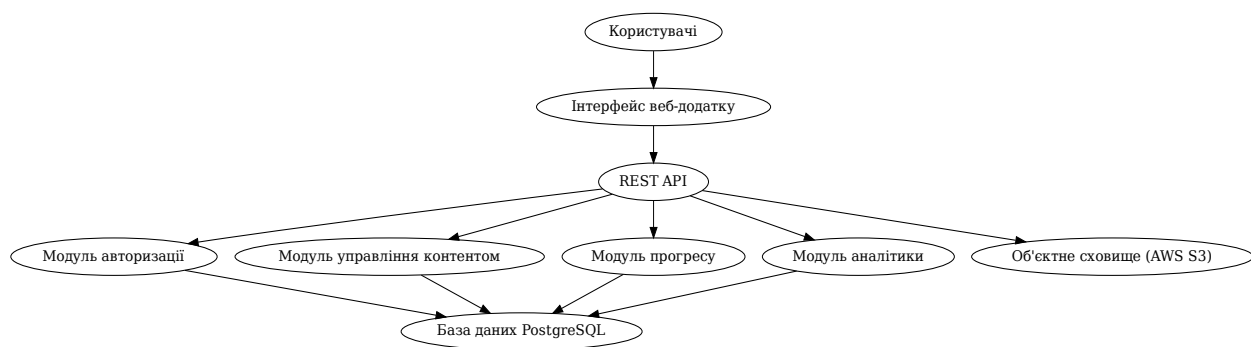


Рис. 2.10. Архітектура бази даних веб-сервісу

Система також підтримує тестове та продуктивне середовище з окремими базами даних. Це забезпечує безпеку при оновленнях, зменшує ризики пошкодження продуктивних даних.

Інтегровано наступні аспекти і принципи для безпеки та масштабованості:

1. Паролі зберігаються із застосуванням bcrypt;
2. Запити до БД реалізовано з перевіркою прав доступу (role-based access control);
3. Використано HTTPS і токени (JWT) для автентифікації;
4. База даних ізольована в приватній підмережі;
5. Налаштовано регулярне резервне копіювання та реплікацію.

Дуже важливим аспектом є висока гнучкість моделі, тому модель дозволяє розширення:

- Додавання нових ролей (наприклад, ментор);
- Додавання підтипів уроків (тести, статті, відео);
- Інтегрування нових типів даних.

Це дозволяє уникнути рефакторингу всієї бази при масштабуванні або зміні логіки сервісу.

## 2.6. Підсумкова оцінка концепції, технологій та обґрунтування обраних рішень

Розробка веб-сервісу для онлайн-курсів є складним багатоаспектним завданням, що потребує інтеграції технічного, управлінського, економічного підходів. У цьому підрозділі здійснюється підсумкова аналітична оцінка концепції проєкту, технічної архітектури, ризиків, сценаріїв впровадження, а

також представлено поглиблену математичну модель взаємодії користувачів із сервісом, що є ключовою складовою для забезпечення якості платформи. Особлива увага приділяється формалізації ризиків, побудові економічного прогнозу та застосуванню теорії черг для аналізу продуктивності системи.

При аналізі ризиків в умовах високої конкуренції в сфері EdTech і технологічної складності платформи ідентифіковано низку критичних ризиків, класифікованих за джерелом походження: технічні, організаційні, фінансові, ринкові. Для кожного з них розроблено стратегії зниження впливу, що подано в табличній формі.

*Таблиця 2.9.*

### **Ризики проекту та стратегії мінімізації**

<b>Тип ризику</b>	<b>Опис ризику</b>	<b>Стратегія мінімізації</b>
Низька початкова залученість користувачів	Користувачі не залишаються на платформі після першого відвідування	Впровадження гейміфікації, персоналізації, підтримка спільноти
Перевантаження серверів при зростанні трафіку	Велика кількість одночасних користувачів спричиняє збої	Горизонтальне масштабування, кешування, балансування навантаження
Порушення безпеки даних	Можливі витоки персональних даних, злам акаунтів	Використання HTTPS, JWT, двофакторна аутентифікація, шифрування даних
Недостатня кількість навчального контенту на старті	Відсутність достатньої кількості якісних курсів	Партнерство з викладачами, запуск MVP з базовим контентом

Продовження табл. 2.9.

Нестабільність фінансування	Брак інвесторів або невиконання бюджету	Мінімізація витрат, фокус на freemium-моделі, пошук грантів
Низька якість користувацького досвіду (UX)	Платформою незручно користуватися, високий рівень відтоку	Прототипування, тестування UX, врахування фідбеку фокус-груп

До найвірогідніших ризиків належить низька початкова залученість користувачів, що нейтралізується введенням персоналізації, гейміфікації та активної email-підтримки. Технічні ризики вирішуються через побудову багаторівневої архітектури, резервне копіювання та тестове середовище. Фінансові виклики компенсуються поступовим масштабуванням та freemium-моделлю монетизації.

З метою оцінки внутрішніх сильних і слабких сторін, а також зовнішніх можливостей і загроз проведено SWOT-аналіз. Він допомагає визначити стратегічну позицію проекту на ринку та сформувані обґрунтовані напрямки розвитку.

Таблиця 2.10.

### SWOT-аналіз проекту веб-сервісу онлайн-курсів

<b>Strengths</b> (Сильні сторони)	<b>Weaknesses</b> (Слабкі сторони)	<b>Opportunities</b> (Можливості)	<b>Threats</b> (Загрози)
Повна україномовна локалізація	Невеликий стартовий бюджет	Зростання попиту на EdTech в Україні	Висока конкуренція з боку глобальних платформ
AI-персоналізація навчання	Відсутність впізнаваності на ринку	Партнерства з університетами	Кібератаки та загрози безпеці даних

Інтуїтивний та доступний інтерфейс	Залежність від грантового фінансування	Державні та міжнародні освітні програми	Макроекономічна нестабільність
------------------------------------	--	---	--------------------------------

Серед сильних сторін – україномовна локалізація, адаптивний дизайн, AI-рекомендації, аналітика прогресу. Слабкими сторонами на старті є обмежене фінансування та відсутність широкої впізнаваності. Як можливості вбачаються співпраця з державними установами та грантовими програмами, а серед загроз – кібербезпека та домінування великих гравців (Coursera, Prometheus).

З огляду на специфіку українського ринку онлайн-освіти, доцільно врахувати приклади з практики локальних EdTech-платформ. Зокрема, Prometheus, як національний гравець, зумів залучити широке коло користувачів за рахунок безоплатного доступу до курсів українською мовою, що свідчить про ефективність моделі співпраці з університетами та державними структурами. Ще одним прикладом є платформа «Дія.Освіта», яка активно інтегрує державні ініціативи у цифрову освіту, підтверджуючи потенціал синергії між EdTech і публічним сектором. Водночас досвід освітніх стартапів, які припинили діяльність через нестачу венчурного фінансування (наприклад, StudyDive), демонструє високу залежність від грантів і ризику нестійкого розвитку. Таким чином, SWOT-аналіз варто інтерпретувати з урахуванням українського контексту, де політична воля та економічна ситуація суттєво впливають на життєздатність EdTech-проектів.

Для перевірки життєздатності сервісу здійснено моделювання витрат і доходів за період у 5 років. У розрахунках враховано витрати на розробку MVP, серверні ресурси, маркетинг, а також прогнозовані доходи від підписок та сертифікацій.

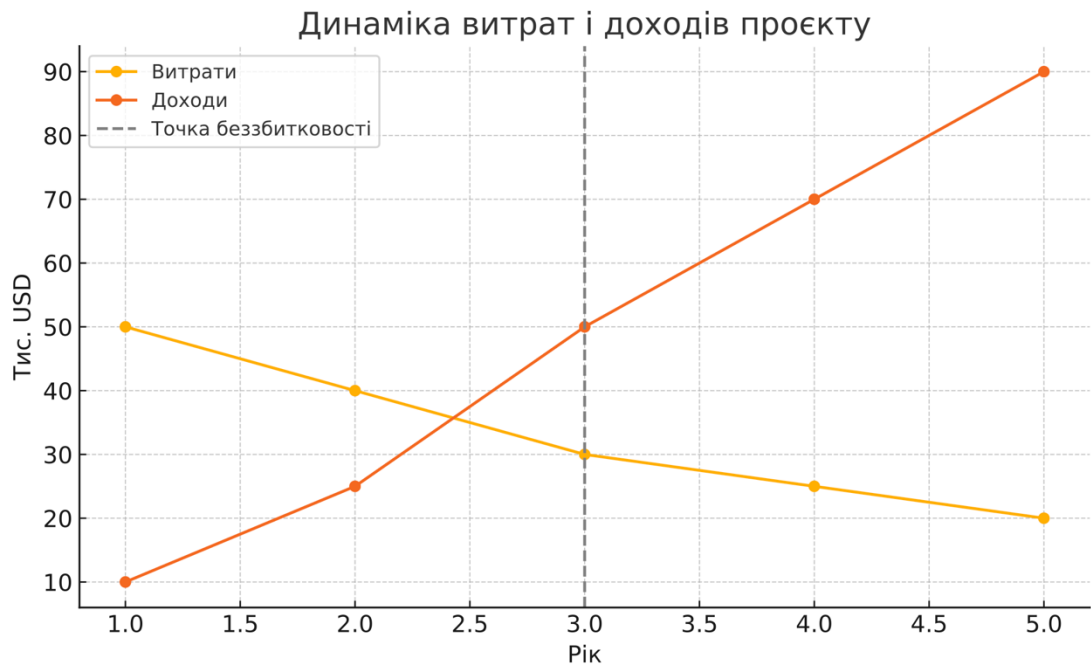


Рис. 2.11. Динаміка витрат і доходів проекту

Згідно з графіком, проєкт досягає точки беззбитковості на третьому році. Після фази високих стартових витрат очікується стабілізація, а зростання прибутку – завдяки масштабуванню користувачів та збільшенню середнього чеку. Така крива характерна для freemium-моделей із органічним зростанням.

Для мінімізації технічних ризиків впроваджено дворівневу схему впровадження: спочатку – внутрішнє MVP-середовище, далі – обмежений реліз і лише потім – масштабоване розгортання на production-середовищі.

Таблиця 2.11.

### Етапи тестування та впровадження веб-сервісу

Етап	Опис	Ціль
<b>Юніт-тестування</b>	Перевірка окремих компонентів коду на коректність	Виявлення помилок у логіці окремих функцій
<b>Інтеграційне тестування</b>	Тестування взаємодії між модулями системи	Забезпечення коректної взаємодії між компонентами
<b>Бета-тестування</b>	Тестування платформи реальними користувачами з обмеженим функціоналом	Збір зворотного зв'язку та виявлення критичних багів

<b>Пілотний запуск</b>	Запуск MVP у контрольному середовищі для обраної аудиторії	Оцінка продуктивності та користувацького досвіду
<b>Повноцінне впровадження</b>	Розгортання сервісу для широкої публіки	Забезпечення масштабованої та стабільної роботи системи

Передбачено проведення юзабіліті-тестування, фокус-груп, інтеграційних випробувань та моніторингу в режимі реального часу. Це дозволяє ітеративно вдосконалювати систему, коригуючи функції ще до повного запуску.

Найважливішим аспектом технічної стійкості платформи є здатність обробляти великий потік користувацьких HTTP-запитів із прийнятним часом відгуку. Для цього побудовано математичну модель черги обслуговування, яка інтерпретує веб-сервер як систему масового обслуговування типу M/M/1.

**Вхідні параметри:**

- $\lambda$  — середня інтенсивність запитів (запитів/сек);
- $\mu$  — продуктивність сервера (запитів/сек);
- $\rho = \lambda / \mu$  — коефіцієнт завантаження.

**Метрики:**

- Середній час в системі:  $W = 1 / (\mu - \lambda)$
- Середній час в черзі:  $W_{queue} = \rho / (\mu - \lambda)$
- Середня кількість запитів у черзі:  $L_{queue} = \rho^2 / (1 - \rho)$

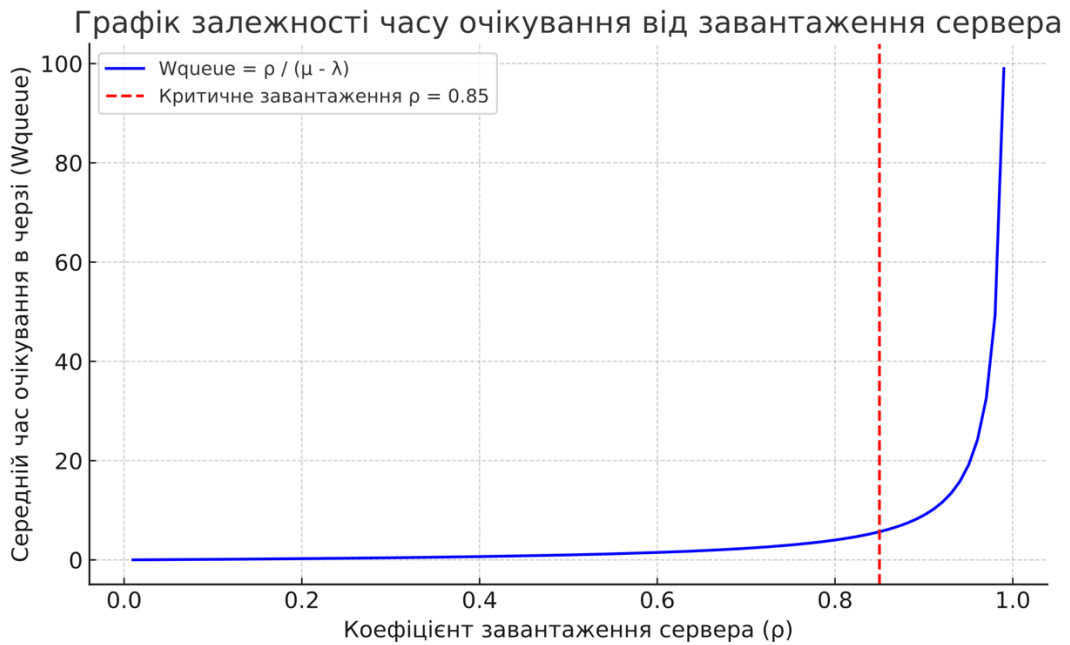


Рис. 2.12. Графік залежності часу очікування від завантаження сервера

Як видно з графіка, при  $\rho > 0.85$  час очікування зростає експоненціально. Це означає, що для забезпечення стабільної роботи системи слід проектувати архітектуру з запасом потужності та цільовим завантаженням не вище 70–80%, особливо в періоди пікового трафіку.

Практичне застосування моделі:

При  $1/\mu = 0.1$  с (один запит обробляється за 0.1 секунди), цільовий час очікування  $W_{total} < 2$  сек.  $\rightarrow \rho \approx 0.95$ . Це означає, що один сервер здатен обслуговувати  $\sim 10$  запитів/сек. За прогнозованої активності 100+ користувачів, необхідне горизонтальне масштабування із застосуванням балансувальника навантаження (наприклад, Nginx) або перехід до M/M/c-моделі для розподілених середовищ. Подібне моделювання широко використовується в інфраструктурах EdTech, SaaS та fintech-системах, де час відповіді безпосередньо впливає на задоволеність користувачів. Модель є частиною процесів «Estimate Activity Resources», «Plan Risk Responses» та «Develop Schedule» згідно з PMBoK. Вона дає змогу формалізувати критерії SLA, обґрунтувати архітектурні рішення та забезпечити безперервну якість обслуговування.

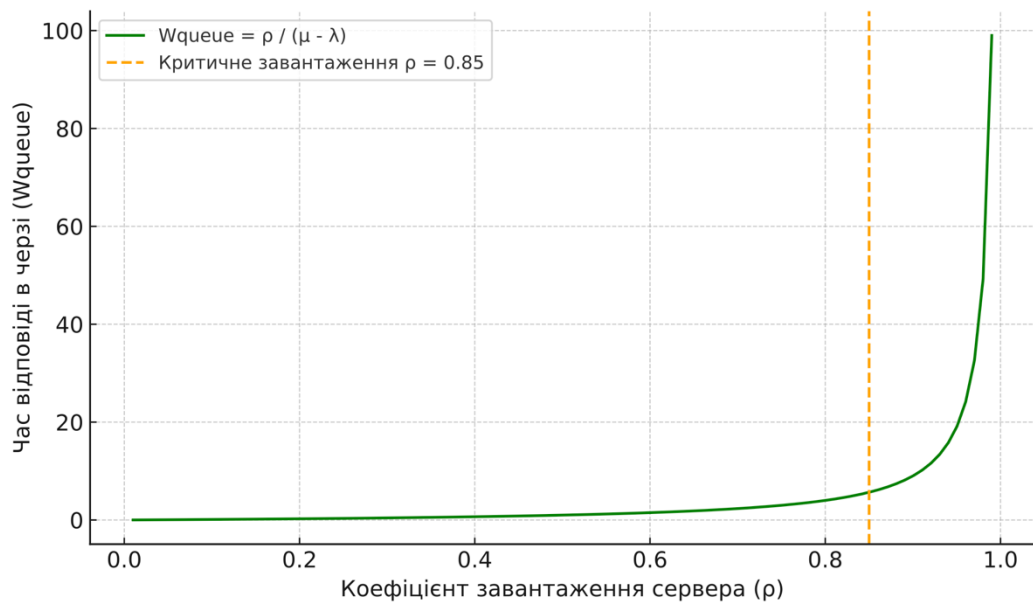


Рис 2.13. Залежність часу відповіді від завантаження сервера ( $W_{queue}$  vs  $\rho$ )

Крім того, модель інтегрується в стратегічне планування. Експоненційна модель зростання користувачів дозволяє прогнозувати навантаження та визначати дату досягнення точки беззбитковості.

Також модель зростання користувачів, наведена раніше:

$$(U(t) = U_0 \cdot e^{kt})$$

– інтегрується в стратегічне планування: на її основі розраховується дата виходу на точку беззбитковості.

Підсумковий аналіз проєкту демонструє системну узгодженість технічної архітектури, економічної моделі та управлінських підходів. Застосування математичних моделей дозволяє не лише прогнозувати розвиток системи, а й формалізувати її поведінку для цілей SLA, масштабування, управління ризиками. Завдяки поєднанню інженерного аналізу, бізнес-планування та наукових підходів, веб-сервіс готовий до масштабованого та якісного запуску в умовах високої конкуренції.

## **Розділ 3. Управління структурою, командою та ресурсами проєкту створення веб-сервісу для онлайн-курсів**

### **3.1. Формулювання мети, концепції та стратегічного бачення проєкту**

Управління IT-проєктами, особливо в сфері EdTech, вимагає чіткої постановки мети, комплексного бачення розвитку продукту та глибокого розуміння його структури. Без цього неможливо забезпечити послідовне планування, ефективне використання ресурсів, раціональне управління ризиками й відповідність очікуванням користувачів. Проєкт має на меті створення інноваційної веб-платформи онлайн-курсів, яка забезпечить широкому колу користувачів доступ до якісної освіти в зручному форматі. Головна мета проєкту – це запуск веб-сервісу, що надасть можливість проходження онлайн-курсів з різних дисциплін з інтерактивними матеріалами та сертифікацією. Відповідно до стандартів PMI, візія проєкту повинна стисло визначати його кінцеву мету та очікуваний результат, об'єднуючи команду навколо спільного бачення [11]. У даному випадку стратегічне бачення полягає у створенні провідної української платформи масових відкритих онлайн-курсів, здатної конкурувати з глобальними аналогами, сприяючи концепції безперервного навчання протягом життя та цифровізації освіти.

Концепція проєкту полягає у поєднанні сучасних технологій веб-розробки з педагогічним баченням онлайн-навчання. Результатом реалізації стане веб-сайт, через який користувачі зможуть реєструватися, обирати курси, переглядати відеолекції, виконувати тести і завдання, а по завершенню – отримувати електронні сертифікати. Проєкт орієнтований і на студентів, фахівців, що прагнуть підвищити кваліфікацію та на освітні установи та бізнеси, які можуть розміщувати власні навчальні програми. Стратегічне бачення розвитку – протягом перших 2 років зайняти значну частку національного ринку EdTech та досягти самоокупності, залучивши тисячі активних користувачів, а в довгостроковій перспективі масштабувати сервіс на міжнародний рівень.

Стратегічні цілі проєкту впливають з основної мети і задають напрямки роботи. Розробити технологічну платформу, наповнити її якісним контентом, залучити й утримати аудиторію, а також забезпечити стійке функціонування сервісу – це основні стратегічні цілі. На рисунку 3.1. представлено дерево цілей проєкту, що демонструє взаємозв'язок головної мети та підцілей, необхідних для її досягнення. Така ієрархічна декомпозиція цілей дозволяє структурувати бачення проєкту та визначити пріоритети.



Рис. 3.1. Дерево цілей проєкту: головна мета та ключові підцілі для її досягнення.

Стратегія досягнення результатів ґрунтується на реалізації зазначених підцілей. Буде реалізовано MVP платформи з базовим функціоналом: особисті кабінети, каталог курсів, відеострімінг лекцій, система тестування та видачі сертифікатів, що є першим продуктовим результатом. Буде налагоджено співпрацю з провайдерами контенту – експертами та викладачами для створення курсів, що дозволить наповнити платформу якісними освітніми матеріалами. Планується розробка плану маркетингу і залучення користувачів: рекламні кампанії у соцмережах, партнерство з університетами та ІТ-компаніями, стимулювання залученості учнів. Буде опрацьовано бізнес-модель проєкту, якою є частково безкоштовний доступ з платними сертифікатами чи розширеними можливостями, пошук інвесторів та грантів. Це забезпечить фінансову стійкість і масштабованість сервісу. Стратегічне бачення полягає у формуванні цілісної екосистеми онлайн-освіти, що включає саму контент-платформу, закриті корпоративні модулі, модулі сертифікації, інструменти аналітики.

Для формалізації архітектури майбутнього сервісу використано метод інтелектуального картографування. Це дозволило структурувати ключові напрями роботи, розбивши проєкт на окремі логічні гілки.

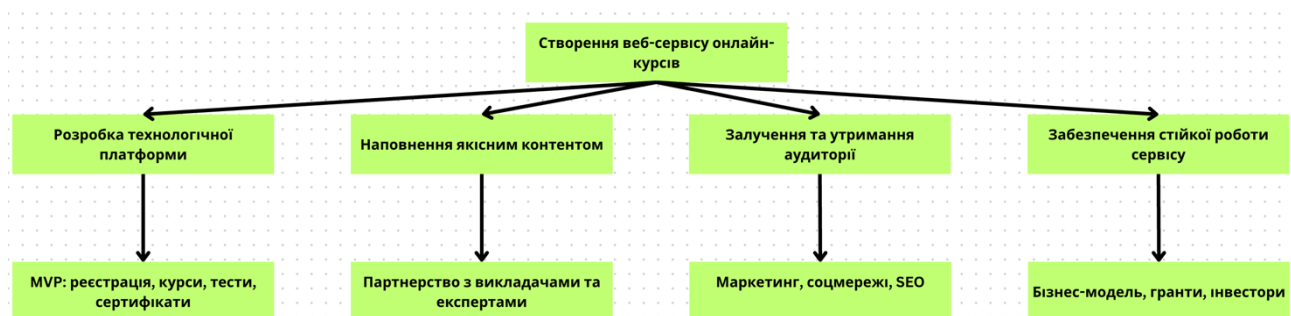


Рис. 3.2. Інтелектуальна карта проєкту створення веб-сервісу онлайн-курсів

Таким чином, було сформовано цілісну модель управління проєктом веб-сервісу онлайн-курсів, що базується на синергії стратегічного планування, ресурсного аналізу, часової координації та фінансової передбачуваності. Застосовані інструменти управління відповідають рекомендаціям стандартів РМВОК та практикам Agile, що забезпечує ґрунтовну основу для подальшого успішного впровадження проєкту.

### 3.2. Аналіз конкурентного та внутрішнього середовища

Перед початком активної реалізації будь-якого ІТ-проєкту надзвичайно важливо здійснити глибокий системний аналіз зовнішнього конкурентного поля та внутрішнього середовища організації або команди. Такий підхід дозволяє не лише оцінити ринкові ризики, визначити потенційних супротивників і сформуванати обґрунтоване позиціонування, але й виявити внутрішні ресурси, які можуть стати джерелом стійкої переваги або, навпаки, точкою вразливості проєкту.

Таблиця 3.1 наводить порівняльну характеристику основних конкурентів нашого проєкту – двох міжнародних платформ та однієї української, що представляють найбільш релевантне зовнішнє середовище.

## Порівняльна характеристика основних конкурентів.

Платформа	Опис і фокус	Масштаб	Модель
<b>Coursera</b>	Глобальна платформа МВОК, курси від університетів і корпорацій; акцент на академічну освіту та сертифікацію	~142 млн користувачів (2023); тисячі курсів і спеціалізацій	Freemium; підписки Coursera Plus
<b>Udemy</b>	Глобальний маркетплейс онлайн-курсів від незалежних інструкторів на різні тематики (ІТ, бізнес, хобі тощо)	>57 млн користувачів (2022); ~213 000 курсів від 70 000+ інструкторів	Платна модель per-course; корпоративні підписки
<b>Prometheus</b>	Найбільша українська платформа безкоштовних масових онлайн-курсів; курси від університетів та експертів українською мовою	Сотні тисяч зареєстрованих слухачів; десятки курсів	Грантова підтримка, донати користувачів

Як видно з таблиці 3.1., міжнародні платформи мають величезну базу користувачів і контенту. Це створює високу конкуренцію на глобальному рівні. Prometheus та інші вітчизняні проекти пропонують контент державною мовою і враховують специфіку національної освіти. Наша платформа буде конкурувати з ними якістю контенту та технологічністю.

Попит на онлайн-курси в Україні зростає, особливо в умовах війни, коли для багатьох дистанційне навчання стало єдиним способом отримання освіти. Це означає збільшення потенційної аудиторії нашого сервісу. Також держава сприяє розвитку EdTech-сектору: спрощується ліцензування, стимулюється інтеграція результатів неформальної освіти тощо.

Аналіз внутрішнього середовища охоплює фактори всередині організації-ініціатора проєкту, що можуть вплинути на успіх. Внутрішнє середовище включає культуру та структуру організації, кадри, фінанси, технології, процеси та накопичений досвід. У нашому випадку проєкт реалізується невеликою ІТ-командою стартапу.

Ключова сила – це досвідчена команда розробників, яку очолює технічний лідер з успішним досвідом запуску подібних продуктів. Є фахівці з UX/UI дизайну та проджект-менеджер із сертифікацією PMP. Команда невелика і гнучка, що сприяє швидкому прийняттю рішень. Використання сучасного стеку технологій та хмарної інфраструктури дає перевагу в швидкості розробки і масштабуванні. Команда відкрита до Agile-методологій, що є сильною стороною у динамічних проєктах.

Слабке місце – відсутність власних освітніх матеріалів на початку. Потрібно залучати викладачів та партнерів для створення контенту, що може уповільнити старт. Також команда стартапу має обмежений досвід саме в педагогічному дизайні, тож потрібні зовнішні консультанти.

Проєкт фінансується за рахунок власних коштів засновників та гранту на підтримку EdTech. Бюджет обмежений, що є слабкістю, тобто необхідно обережно планувати витрати і, можливо, шукати додаткові інвестиції.

Організаційна структура відносно плоска, з високим рівнем взаємодії. Невеликий масштаб дозволяє працювати як єдина проєктна команда без бюрократії. Це підвищує гнучкість, але може бути ризиком у питаннях стабільності.

У компанії впроваджені Agile-підходи управління проєктами (Scrum), і це сприяє самоорганізації та залученню команди. Є культура відкритого

обговорення і постійного вдосконалення процесів, що буде підтримувати мотивацію учасників.

Для узагальнення зовнішніх і внутрішніх факторів застосуємо SWOT-аналіз (рис. 3.3.). Це інструмент стратегічного планування, що дозволяє глибоко зрозуміти внутрішні ресурси організації та їх значення у досягненні конкурентних переваг.

<b>Strengths:</b> <ul style="list-style-type: none"><li>• Досвідчена технічна команда</li><li>• Гнучкість стратегії</li><li>• Сучасні технології</li><li>• Agile-культура</li></ul>	<b>Weaknesses:</b> <ul style="list-style-type: none"><li>• Брак контенту на старті</li><li>• Обмежене фінансування</li><li>• Недостатній досвід у педдизайні</li></ul>
<b>Opportunities:</b> <ul style="list-style-type: none"><li>• Зростання EdTech в Україні</li><li>• Партнерства з університетами</li><li>• Підтримка держави</li></ul>	<b>Threats:</b> <ul style="list-style-type: none"><li>• Конкуренція з глобальними платформами</li><li>• Ризик фінансування</li><li>• Кіберзагрози</li></ul>

Рис. 3.3. SWOT-аналіз проєкту

SWOT-аналіз на цьому етапі стосується менеджменту проєкту в цілому, а не лише розроблюваного продукту. Він доповнює попередній аналіз з табл. 2.10, деталізуючи специфіку викликів для команди, термінів, ресурсів і реалізації.

Аналіз показує, що проєкт спирається на потужну команду та технології, але має подолати внутрішні слабкості, як-то брак контенту та коштів.

Водночас ринковий попит і партнерства створюють умови для зростання, якщо ефективно ними скористатися. Не слід нехтувати і загрозами, бо доведеться активно диференціювати продукт від глобальних платформ, а також залучати фінансування для розвитку.

Проєкт реалізується командою, яка має такі основні характеристики представлені в таблиці 3.2.

## Вимоги до характеристики членів команди

Напрямок	Кваліфікація
<b>Технічні спеціалісти</b>	Досвід в архітектурі веб-додатків, базах даних, frontend/backend розробці
<b>UI/UX-дизайнери</b>	З практикою реалізації освітніх та комерційних інтерфейсів
<b>Data-аналітики</b>	Знання принципів побудови освітньої аналітики та її візуалізації
<b>Project Manager</b>	Із досвідом координації SCRUM-процесів та впровадження WBS у проєктах середнього масштабу
<b>Business Analyst</b>	З досвідом роботи з освітніми стартапами

Попри відносну “молодість” команди, внутрішня культура проєкту має характеризуватися рисами прозорості у прийнятті рішень, пріоритетом якості над швидкістю, схильністю до постійного навчання, зворотного зв’язку та горизонтальним управлінням.

Для глибшого розуміння потенційних причин проблем та бар’єрів реалізації проєкту застосовано метод дерева причин та наслідків (рис. 3.4.). Це дозволило структурувати можливі джерела складнощів у технічних категоріях, маркетингових категоріях, категорії UX-дизайну та зовнішніх впливах.

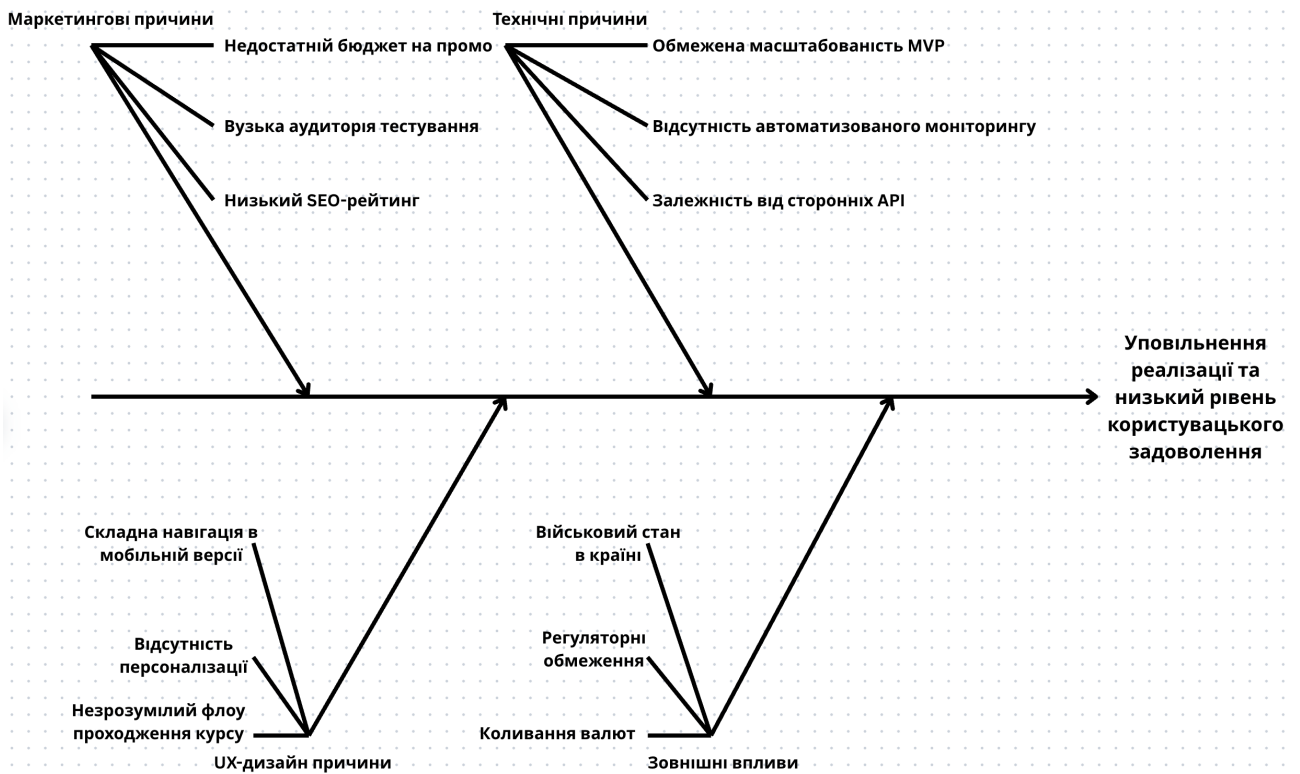


Рис. 3.4. Діаграма Ісікави

Виявлені ризики пов'язані з браком початкового контенту та обмеженим бюджетом, компенсуються гнучкою структурою управління та високою мотивацією учасників.

### 3.3. Дерево цілей проєкту та стратегія досягнення результатів

Основною метою проєкту, як зазначалося раніше, є покращення доступу до онлайн-освіти в Україні шляхом створення адаптивного, персоналізованого та зручного веб-сервісу. Ця мета є комплексною, адже вона включає технологічні та соціальні аспекти.

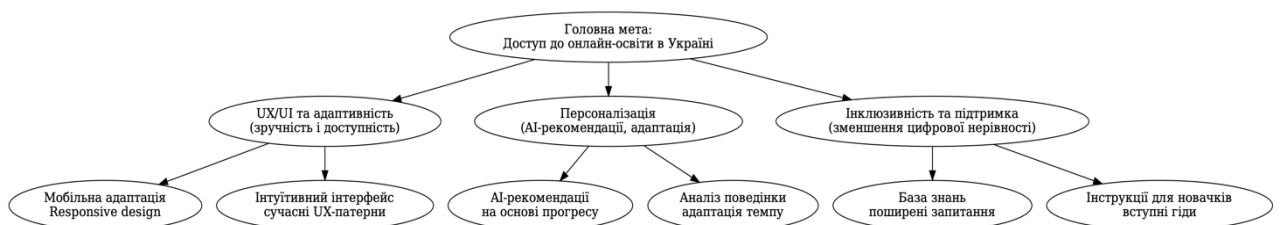


Рис. 3.5. Дерево цілей проєкту веб-сервісу для онлайн-курсів

Ця гілка передбачає зосередження на трьох ключових компонентах розробки UX/UI-інтерфейсу, що відповідає сучасним стандартам інтуїтивної взаємодії, реалізації мобільної адаптації, включно з responsive-дизайном і мобільними web-first патернами, інтеграції системи управління курсами, що

дозволяє викладачам створювати модулі, керувати матеріалами, додавати тести, відслідковувати активність. Очікуваним результатом цього блоку є підвищення показника задоволення користувачів та залучення аудиторії (engagement rate).

Персоналізація розглядається як один з найважливіших факторів конкурентоспроможності сучасних EdTech-продуктів. У межах проєкту реалізується впровадження систем рекомендацій на основі AI, які аналізують прогрес студента та пропонують відповідні наступні курси або матеріали, аналіз поведінки користувачів, що дозволяє системі адаптувати темп і рівень складності матеріалу до особливостей кожного слухача.

Цей напрям має на меті зменшення цифрової нерівності, що особливо актуально в українському контексті, зважаючи на неоднорідність технічної підготовки населення. Він охоплює розробку вступних інструкцій та створення бази знань, де акумулюються відповіді на поширені запитання, технічні інструкції.

Створене дерево цілей проєкту не лише формує стратегічну основу для побудови WBS-структури та беклогу, але й дозволяє чітко візуалізувати життєвий цикл (ЖЦ) реалізації веб-сервісу. На рисунку 3.6 подано модель ЖЦ проєкту, яка охоплює основні фази: ініціалізацію та планування, розробку MVP, інтеграцію та тестування, маркетинг і запуск, а також подальшу підтримку продукту. Така структуризація дозволяє зіставити задачі з відповідальними ролями в команді, забезпечити прогнозоване використання ресурсів і організувати поетапне досягнення цілей.

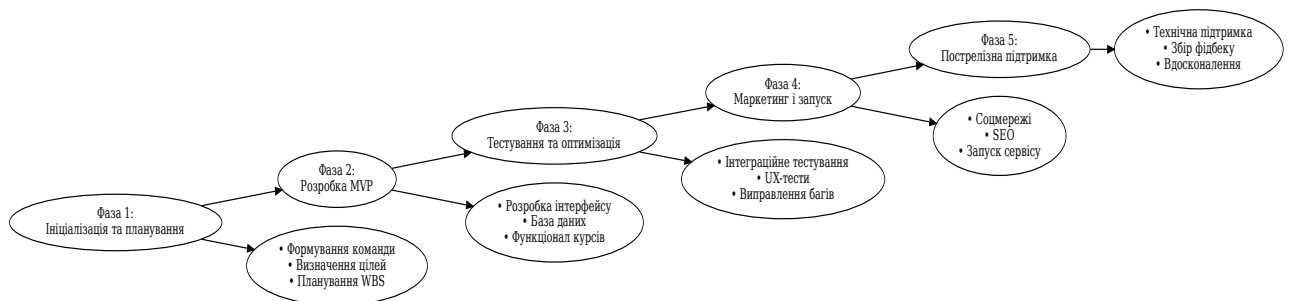


Рис. 3.6. Життєвий цикл проєкту онлайн-курсів

Попри візуальну подібність до каскадної моделі, дана схема (рис. 3.6) відображає узагальнену послідовність етапів реалізації, притаманну будь-якому

ІТ-проєкту. Водночас у рамках кожного з цих етапів (зокрема, реалізації, тестування, вдосконалення) застосовуються гнучкі методи управління – ітеративна розробка, спринти, зворотний зв'язок, адаптація. Таким чином, життєвий цикл є структурованим за фазами, але виконується за підходом Agile, із гнучкою організацією роботи всередині етапів.

Обрана модель управління ґрунтується на гнучкій методології Agile, зокрема підході SCRUM, який найкраще відповідає природі даного ІТ-проєкту. SCRUM дозволяє реалізовувати функціональність у вигляді ітеративних інкрементів, швидко реагувати на зміну вимог, активно залучати замовника на всіх етапах і мінімізувати ризики. Класичні підходи, як каскадна модель (Waterfall) чи PMBOK, передбачають жорстке фазове планування без можливості швидкого коригування, що є малоефективним у контексті стартапів або EdTech-проєктів. Відповідно, впровадження SCRUM у комбінації з поетапною реалізацією ЖЦ забезпечує високу адаптивність, прозорість і результативність управління проєктом.

### **3.4. Управління змістом проєкту**

Управління змістом проєкту (Scope Management) забезпечує контроль всіх робіт для створення продукту проєкту. Згідно з PMBOK, ключовим інструментом є ієрархічна декомпозиція робіт (WBS), яка поділяє весь обсяг проєкту на структуровані компоненти [12]. На основі вже визначених цілей виконаємо декомпозицію проєкту на підпроєкти, етапи та конкретні завдання.

Декомпозиція всього проєкту на основі його життєвого циклу є одним із найважливіших аспектів успішного управління проєктом. Побудована WBS-структура проєкту подана на рисунку 3.7. Вона реалізована у вигляді багаторівневої ієрархії завдань, що охоплює всі ключові фази життєвого циклу веб-сервісу — від ініціалізації до підтримки після запуску. Третій рівень деталізації дозволяє більш точно координувати дії команди, ідентифікувати залежності між модулями та ефективно керувати ресурсами.

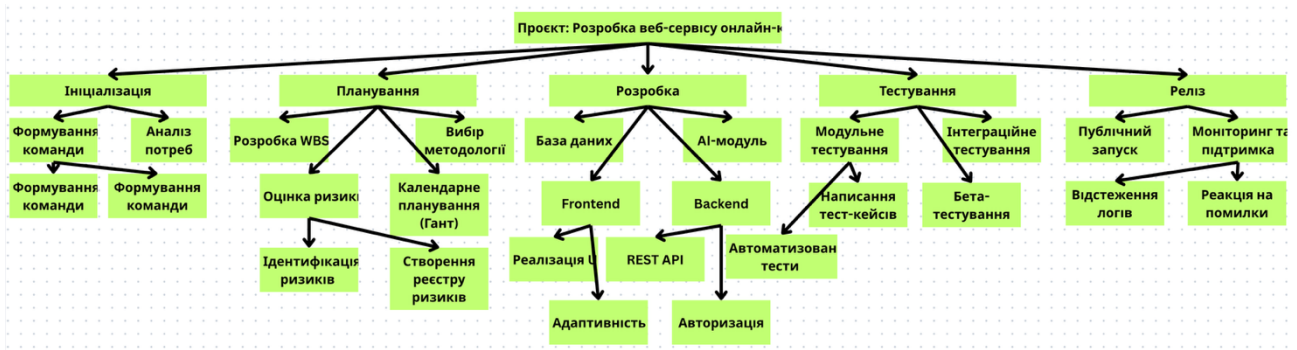


Рис. 3.7. Ієрархічна WBS-структура проєкту розробки веб-сервісу онлайн-курсів

Цей формат дає змогу системно керувати календарним планом, ідентифікувати точки контролю (milestones), будувати критичний шлях та адаптувати структуру до SCRUM-логіки спринтів.

Наступний рівень деталізації – це WBS, побудований за функціональними модулями майбутнього веб-сервісу (рис. 3.8.).



Рис. 3.8. WBS проєкту за модулями продукту

Ця структура дозволяє розробникам і менеджерам розуміти технологічну архітектуру проєкту, формувати беклог відповідно до функціональних залежностей.

На основі попередніх WBS-структур сформуємо беклог проєкту для реалізації мінімально життєздатного продукту.

Таблиця 3.3.

**Беклог проєкту: список ключових завдань**

ID	Назва задачі	Пріоритет	SP	Статус	Відповідальний
1	Реєстрація нового користувача	Високий	3	Backlog	Frontend

Продовження табл. 3.3.

2	Каталог курсів з фільтрами	Високий	5	Backlog	Frontend
3	Інтеграція відеоплеєра	Середній	8	In Progress	Frontend
4	Кабінет викладача	Високий	8	Backlog	PM+Backend
5	Модель бази даних	Високий	8	In Progress	Backend
6	Тести та оцінювання	Високий	5	Backlog	Backend
7	Електронний сертифікат	Середній	8	Backlog	Frontend
8	Аналітика прогресу	Середній	3	Backlog	Data Analyst
9	Адмін-панель модерації	Середній	5	Backlog	Admin
10	Авторизація користувача	Високий	8	Backlog	Backend
11	Редагування профілю	Середній	2	Backlog	Frontend
12	Фільтрація та пошук	Середній	2	Backlog	Frontend
13	Перегляд прогресу	Середній	2	Backlog	Frontend
14	Редагування курсу	Середній	3	Backlog	Backend
15	Додавання відео/тестів	Середній	4	Backlog	Backend
16	Налаштування ролей	Середній	2	Backlog	Admin
17	AI-рекомендації	Середній	4	Backlog	AI Engineer
18	Аналіз поведінки	Середній	3	Backlog	AI Engineer
19	Пошта/нотифікації	Низький	2	Backlog	Marketing
20	Імпорт матеріалів	Низький	2	Backlog	CM

WBS забезпечує структурний “скелет” проекту, тоді як беклог забезпечує динамічний список робіт. Елементи WBS в управлінні перетворюються на завдання, які можна корелювати з юзер-сторі в беклозі. Таким чином, такий підхід дозволяє дотримуватись вимог до повноти планування і одночасно зберігати гнучкість Agile.

### 3.5. Організаційна структура та управління командою проєкту

Успіх проєкту значною мірою залежить від правильної організації команди і ефективного управління ресурсами. Проєкт веб-сервісу онлайн-курсів виконується командою розміром близько 10 осіб, що включає в себе розробників, тестувальників, дизайнера UI/UX, менеджера проєкту, фахівця з контенту.

Сформована окрема проєктна команда, яка повністю зосереджена на цьому продукті. На рисунку 3.9. зображено діаграму команди. Роль керівника проєкту (Project Manager) виконує сертифікований менеджер, відповідальний за координацію роботи та взаємодію зі стейкхолдерами. В рамках Agile-процесу він також виступає в ролі Scrum Master. В команді виокремлено роль власника продукту (Product Owner) – представника замовника, який визначає пріоритети беклогу. Безпосередньо розробкою займається крос-функціональна команда розробки, що включає front-end та back-end розробників, тестувальників і UX-дизайнера.

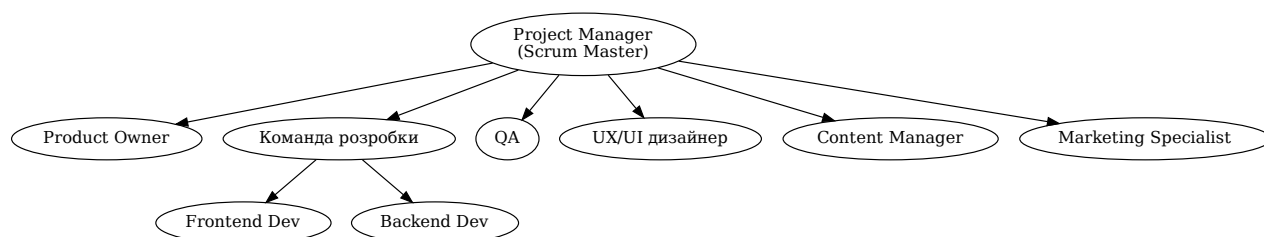


Рис. 3.9. Організаційна структура команди проєкту

Керівник проєкту виконує функції лідера і координатора, але уникає мікроменеджменту, тобто команда сама вирішує, як виконувати поставлені задачі. Така самоорганізація підвищує залученість та відповідальність членів команди.

Ролі та обов'язки команди:

1. Project Manager/Scrum Master: відстеження прогресу, усунення перешкод, комунікація з стейкхолдерами, проведення щоденних стендапів.
2. Product Owner: формування бачення продукту, пріоритизація беклогу, приймання результатів спринтів, збирання зворотного зв'язку від користувачів і бізнесу.

3. Frontend, Backend: розробка функціоналу згідно з беклогом, написання коду.

4. QA: тестування функціоналу, забезпечення відповідності вимогам.

5. UX/UI дизайнер: проєктування інтерфейсу користувача, прототипування.

6. Content manager: взаємодіє з авторами курсів, допомагає структурувати освітній контент.

7. Marketing specialist: просування платформи, ведення соцмереж, організація рекламних кампаній.

Враховуючи стартап-культуру, зроблено акцент на нематеріальній мотивації: залучення у прийняття рішень, гнучкий графік, можливості навчання, атмосфера співпраці. Застосовується принцип Servant Leadership, коли менеджер проєкту допомагає команді максимально реалізувати потенціал, а не просто роздає накази.

Планується запровадити регулярні зустрічі: щоденні 15-хвилинні стендап-мітинги, де обговорюється прогрес і перешкоди. Вся проєктна інформація прозора для команди – використовується спільний таск-трекер (Jira) та репозиторій коду. Це сприяє колективній відповідальності, тому що кожен розуміє загальну картину і свій внесок.

На початку проєкту буде приділено увагу соціальній взаємодії та розвитку команди. Будуть проводитись командні воркшопи для знайомства та визначення спільних цінностей. Як результат, вже до кінця другого спринту команда перейде до ефективної співпраці.

Організаційна структура та стиль управління командою вибрані відповідно до специфіки проєкту. Невеликий інноваційний стартап потребує гнучкої, самовмотивованої команди з мінімумом ієрархії. Представлена Agile-структура дозволяє максимально використовувати творчий і технічний потенціал учасників, забезпечуючи при цьому необхідний рівень контролю та спрямування з боку менеджера проєкту. Це закладає міцну основу для успішного виконання проєкту в задані строки і з очікуваною якістю.

### 3.6. Управління часом проєкту

Управління часом (project time management) є одним із ключових процесів у системі управління IT-проєктами. Його основна мета полягає у забезпеченні своєчасного досягнення поставлених цілей через структуроване планування, чітке визначення тривалості етапів і задач, контроль за виконанням та адаптацію до змін. У контексті розробки веб-сервісу для онлайн-курсів, своєчасність розробки є критичним чинником не лише для відповідності очікуванням аудиторії, але й з огляду на динамічність ринку.

У межах цього проєкту було обрано гібридну модель управління часом, яка поєднує класичне поетапне планування згідно з підходами PMBOK та ISO 21500 із гнучкими ітераційними практиками SCRUM. Такий підхід дозволяє забезпечити передбачуваність і тактичну гнучкість у реалізації функціональності.

Часова модель проєкту розроблялася у декілька етапів:

1. Ідентифікація ключових фаз життєвого циклу проєкту;
2. Розбиття основної роботи на спринти тривалістю два тижні кожен;
3. Побудова пофазного календарного плану у форматі діаграми Ганта;
4. Побудова окремої діаграми для реалізації функціональних модулів;
5. Визначення критичного шляху та буферів часу;
6. Формалізація засобів моніторингу та контролю виконання.

Управління часом здійснюється на двох рівнях: стратегічному (по фазах) і тактичному (через спринти). На стратегічному рівні виділено п'ять основних фаз: ініціалізація, планування, розробка, тестування та реліз. На тактичному рівні реалізація кожного блоку функціоналу розбивається на спринти, які завершуються робочим інкрементом продукту.

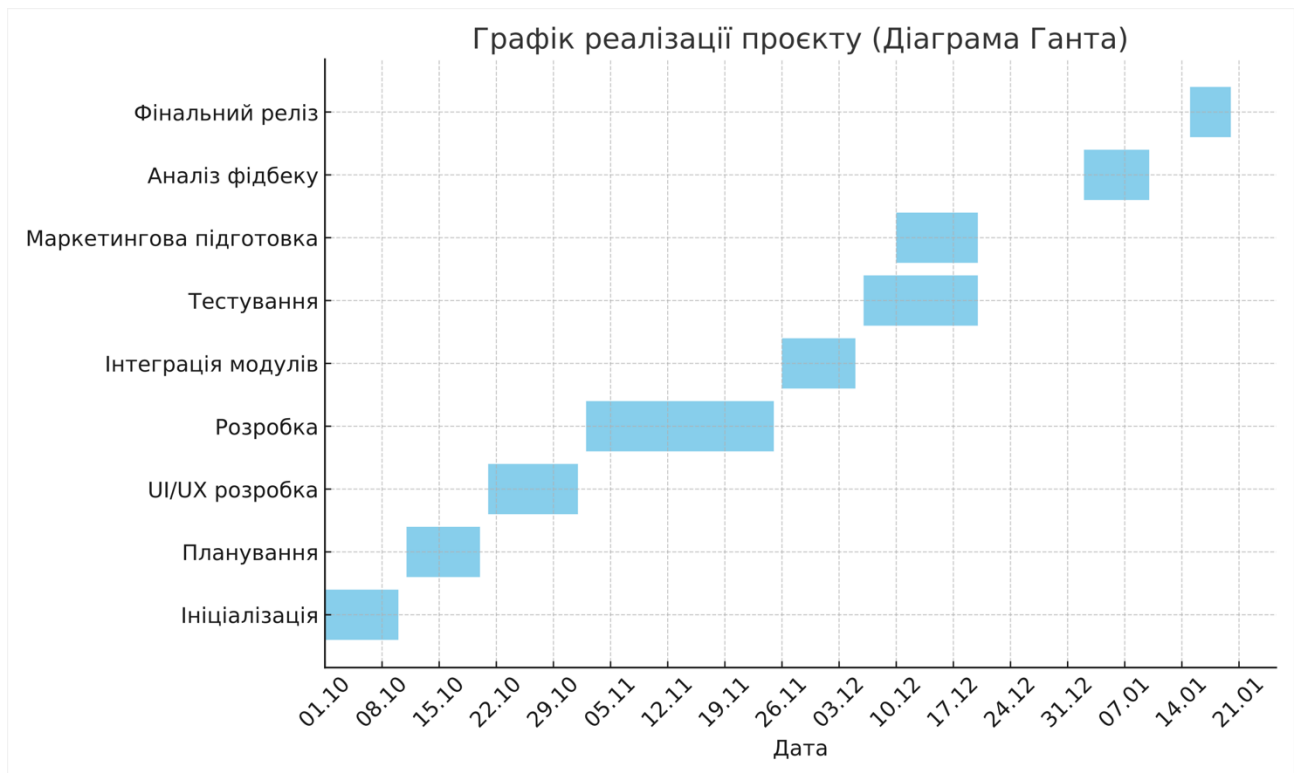


Рис. 3.10. Макет діаграма Ганта – Загальний календарний план проекту

Проект розпочинається з ініціалізації, яка включає формування команди, визначення стейкхолдерів, аналіз потреб, а також узгодження візії продукту. Цей етап триває з 1 по 10 жовтня 2024 року.

Планувальна фаза триває до 20 жовтня і охоплює побудову WBS, складання беклогу, розробку структури команди та попереднє бюджетування. На цьому етапі також створюються діаграми залежностей і карта ризиків.

Основна частина реалізації припадає на період з 21 жовтня до 25 грудня 2024 року. Вона складається з трьох ключових двотижневих спринтів. У межах кожного спринту команда працює за SCRUM-принципами: планування спринту, щоденні стендапи, демонстрація інкременту.

У фазі тестування з грудня 2024 по січень 2025 відбувається інтеграційна перевірка модулів, навантажувальні та UX-тести, оптимізація продуктивності. Паралельно з цим проводиться маркетингова підготовка, підключення служби підтримки, підготовка до релізу.

Фінальна фаза охоплює період з 15 до 20 січня 2025 року і передбачає офіційний запуск продукту на продакшн-середовищі, контрольну перевірку, збір відгуків перших користувачів.

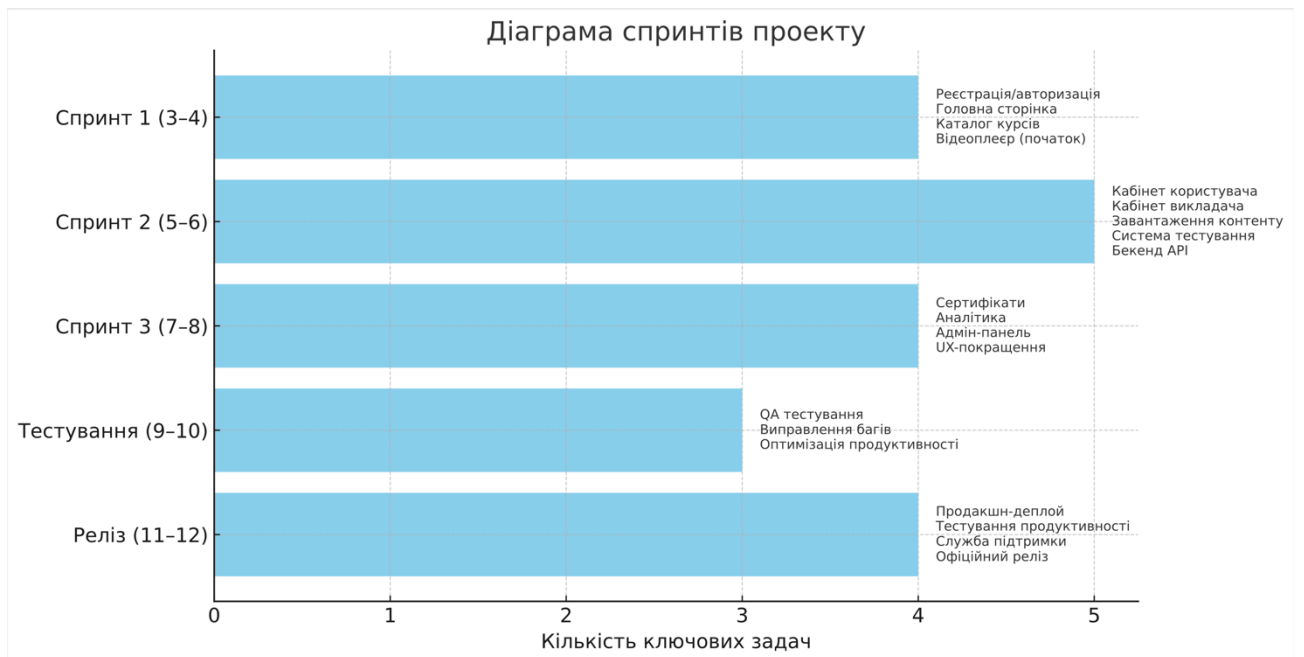


Рис. 3.11. Діаграма спринтів – План реалізації інкрементів за SCRUM

Реалізація функціональності здійснюється по модулях: авторизація, каталог курсів, особисті кабінети, відеоплеєр, система тестування, AI-рекомендації, панель адміністратора, аналітика, система сертифікації. Модульна структура дозволяє розподілити завдання між командами Frontend, Backend, QA, UX/UI та аналітиками, уникнувши дублювання зусиль.

Затримка в будь-якому з наведених напрямів автоматично впливає на весь проектний графік, наш критичний шлях охоплює:

1. Створення інтерфейсу користувача (UI/UX);
2. Розробку серверної логіки (Backend);
3. Тестування взаємодії компонентів;
4. Розгортання та реліз.

Для контролю виконання плану використовуються такі інструменти:

- Jira – відображення статусів задач в режимі реального часу;
- Burn-down chart – контроль темпів виконання спринтів;
- Sprint velocity – облік story points;
- Milestone control – фіксація ключових подій;
- Earned Value Management – оцінка запланованих обсягів до фактично виконаних робіт.

З метою зменшення ризиків до кожного спринту додано буфер у 1-2 дні для виправлення помилок, тестування або неочікуваних завдань. У фінальну фазу закладено п'ятиденний резерв для доопрацювань без зміщення дати релізу.

У разі затримок оцінюється доцільність змін, перерозподіляються завдання або додається додатковий спринт. Усі зміни фіксуються менеджером проєкту та обговорюються із ключовими стейкхолдерами.

### **3.7. Фінансове планування і моделювання ефективності**

Фінансове планування та оцінка ефективності є ключовими аспектами в управлінні IT-проєктами. Для веб-сервісу онлайн-курсів, який реалізується як EdTech-стартап на конкурентному ринку, ці елементи визначають здатність досягати стратегічних цілей.

Мета фінансового моделювання полягає в обґрунтуванні інвестицій, прогнозу доходів і витрат, а також визначенні показників ефективності (ROI, точка беззбитковості), які дозволяють оцінити економічну життєздатність проєкту. Запроваджено модель оцінки успіху, що враховує наступні чинники: рівень залученості користувачів, якість управління, підтримку керівництва та командну компетентність.

Загальний бюджет на стартову фазу реалізації (3 місяці) становить 670 000 грн. Основні статті витрат представлені у таблиці 3.4.:

*Таблиця 3.4.*

#### **Загальний бюджет на стартову фазу реалізації**

<b>Тип витрат</b>	<b>Сума (грн)</b>	<b>Примітка</b>
Заробітна плата команди	450 000	Розробники, дизайнер, QA, PM, аналітик
UI/UX-дизайн	60 000	Прототипи, дизайн-система
Хостинг та інфраструктура	30 000	Сервери, домени, інфраструктурні витрати
Маркетинг	80 000	Соцмережі, SEO, email-кампанії
Інтеграція API та сервісів	20 000	Платіжні шлюзи, сервіси аналітики

Юридичне оформлення	15 000	Реєстрація ТОВ, документи, політика конфіденційності
Резерв непередбачуваних витрат	15 000	Непередбачувані технічні або адміністративні витрати
<b>Разом:</b>	<b>670 000</b>	

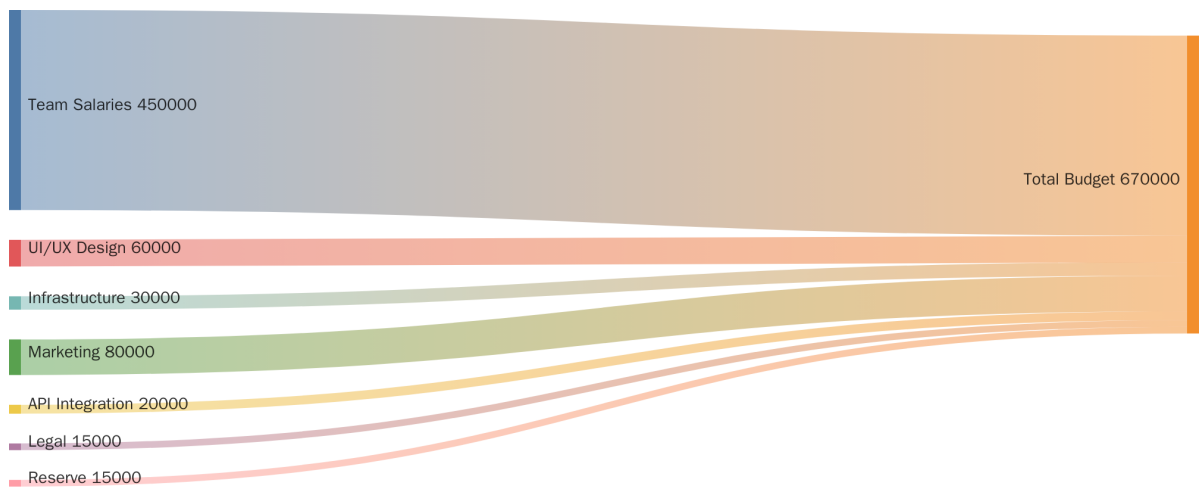


Рис. 3.12. Sankey-візуалізація бюджету проекту за категоріями витрат

Фінансова модель охоплює вісім місяців після запуску проекту. У перші три місяці (фаза розробки) витрати перевищують доходи, що є типовим для MVP у сфері EdTech. Починаючи з січня 2025 року, за рахунок повноцінного запуску функціоналу та активізації маркетингу, очікується зростання прибутковості.

- **Жовтень – грудень 2024:** дефіцитна фаза, витрати ~670 000 грн;
- **Січень – лютий 2025:** початковий дохід, зменшення дефіциту;
- **Березень – травень 2025:** вихід на позитивний грошовий потік.

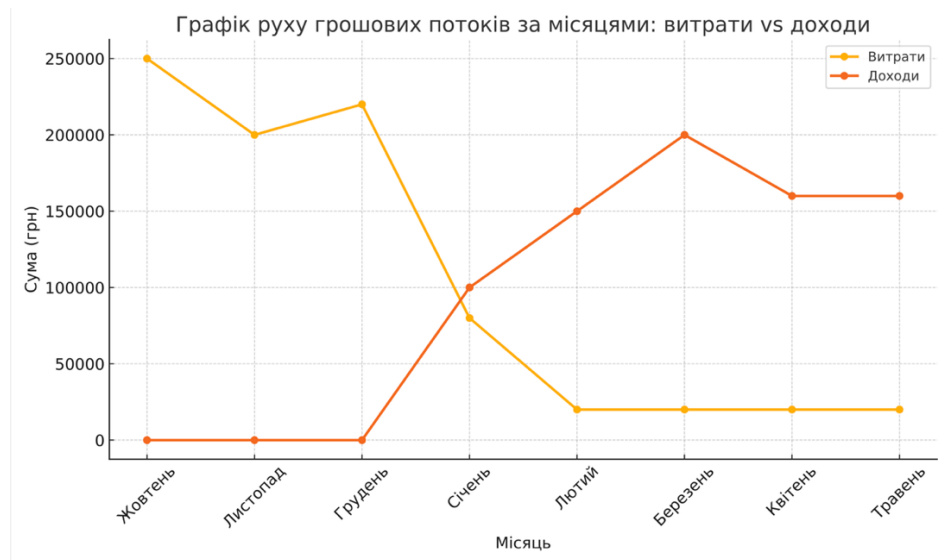


Рис 3.13. Графік руху грошових потоків за місяцями: витрати vs доходи

Проект виходить на точку беззбитковості на початку п'ятого місяця після запуску, тобто в травні 2025 року. З цього моменту доходи починають покривати попередні інвестиції. Визначено також базовий показник ROI:

$$ROI = \frac{\text{Доходи} - \text{Інвестиції}}{\text{Інвестиції}} \times 100\% = \frac{770000 - 670000}{670000} \times 100\% \approx 14.93\%$$

Хоча цей рівень виглядає помірним, він є прийнятним для MVP-етапу, особливо в умовах тестування моделі зростання.

З урахуванням ринкової невизначеності проведено сценарне моделювання:

1. Оптимістичний сценарій: прибуток – 1 500 000 грн; ROI  $\approx$  123%;
2. Реалістичний сценарій: прибуток – 1 100 000 грн; ROI  $\approx$  64%;
3. Песимістичний сценарій: прибуток – 700 000 грн; ROI  $\approx$  4.5%.

Оскільки кожен із наведених фінансових сценаріїв прямо залежить від низки внутрішніх і зовнішніх чинників, додаткову увагу було приділено аналізу ключових ризиків, здатних вплинути на строки, вартість і ефективність реалізації проекту.

Основні ризики класифіковано за джерелом походження: технічні, організаційні, фінансові та ринкові. Для кожного з них проведено якісну оцінку ймовірності настання, рівня впливу на результати, а також розроблено відповідні стратегії реагування. Підсумки аналізу подано у таблиці 3.5.

## Ключові ризики проєкту та стратегії реагування

№	Назва ризику	Тип	Ймовірність	Вплив	Стратегія
1	Затримка у розробці ключових модулів	Технічний	Середня	Високий	Буфер часу, розбиття задач на спринти
2	Відсутність якісного контенту на старті	Організаційний	Висока	Високий	Партнерства з викладачами
3	Нестабільність хостингу або API	Технічний	Низька	Високий	Хмарна інфраструктура з резервуванням
4	Брак фінансування на етапі масштабування	Фінансовий	Середня	Високий	Поступове зростання, гранти
5	Низька залученість користувачів після запуску	Ринковий	Середня	Високий	UX-опитування, гейміфікація

Застосовуються KPI, що дозволяє вимірювати успішність продукту в динаміці:

1. Кількість активних користувачів через 6 місяців (>5000);
2. Відсоток завершення курсів (>45%);
3. Середній рейтинг платформи (>4.2/5);

4. Частка повторних користувачів (>30%);
5. Технічна стабільність (менше 1 багу на 100 користувачів).

KPI регулярно вимірюються та зіставляються з плановими показниками для аналізу відхилень і корекції стратегії.

Для прогнозування потенційної ефективності проєкту запроваджено модель індексу успіху (PSI) на основі п'яти ключових факторів:

1. Залученість користувачів (25%);
2. Чіткість і стабільність вимог (20%);
3. Кваліфікація команди (20%);
4. Підтримка керівництва (20%);
5. Якість початкового планування (15%).

Формула PSI:

$$PSI = 0.25X_1 + 0.20X_2 + 0.20X_3 + 0.20X_4 + 0.15X_5$$

При поточних оцінках ( $X_1-X_5 \approx 0.8-0.95$ ) значення PSI перебуває на рівні  $\sim 0.85-0.9$ , що свідчить про високу ймовірність успіху.

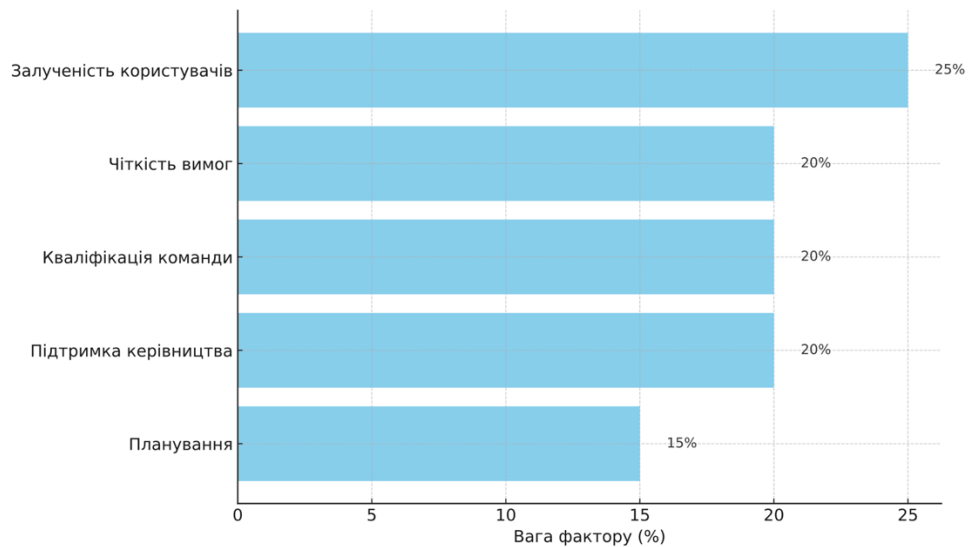


Рис. 3.14. Діаграма ваг факторів успіху у PSI-моделі

PSI використовується як інструмент what-if аналізу. Наприклад, зменшення показника  $X_2$  (вимоги) з 0.9 до 0.6 призводить до зниження PSI на  $\sim 6\%$ , що сигналізує про зростання ризику невдачі. Ця модель дозволяє приймати превентивні управлінські рішення.

## **Розділ 4. Моніторинг, контроль і реалізація проєкту створення веб-сервісу для онлайн-курсів**

### **4.1 Розгортання та впровадження веб-сервісу**

Етап розгортання є критичним у проєкті розробки веб-сервісу для онлайн-курсів, оскільки саме він визначає готовність продукту до взаємодії з цільовою аудиторією та забезпечує надійність, масштабованість і безперебійну роботу сервісу в реальних умовах. В межах цієї фази реалізуються конфігурація середовища, розгортання backend- та frontend-частин, підключення бази даних і CI/CD-процесів, а також налаштування засобів моніторингу та безпеки.

Проєкт реалізовано з урахуванням принципів DevOps та container-based deployment. Основне середовище побудоване за допомогою:

- Docker-контейнерів, які розділяють сервіси на незалежні модулі: API, база даних, frontend;
- CI/CD пайплайну, налаштованого у GitHub Actions із автоматичним деплоєм на staging/production;
- PostgreSQL як основної реляційної бази даних;
- AWS як основного хмарного провайдера.

Під час налаштування інфраструктури особливу увагу було приділено розмежуванню середовищ:

- Development (локальне середовище для розробки з тестовою базою);
- Staging (імітація продакшн-середовища для ручного тестування та перевірки функцій);
- Production (фінальне середовище, доступне користувачам).

На рисунку 4.1. подано архітектуру розгортання веб-сервісу:

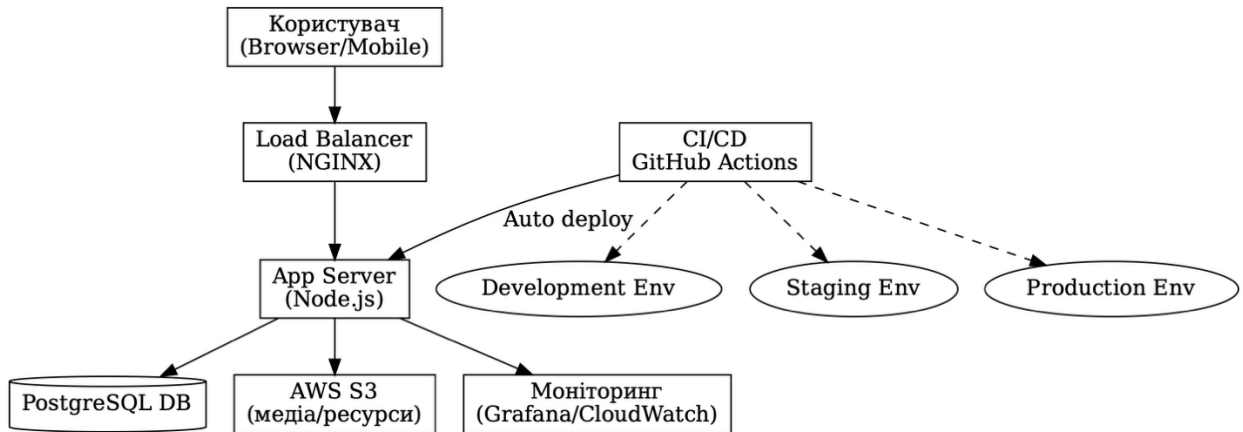


Рис. 4.1. Схема розгортання системи

Ця модель демонструє взаємодію між основними компонентами: користувацький інтерфейс (browser/mobile), балансувальник навантаження (nginx), сервер додатку, база даних, засоби моніторингу, інструменти CI/CD і зберігання медіаконтенту. Такий підхід гарантує гнучкість масштабування, можливість безперервного оновлення без даунтайму та централізовану систему логування.

Розгортання веб-сервісу розпочалося з етапу MVP, що передбачав обмежений набір функцій: реєстрація/логін, список курсів, перегляд одного демонстраційного модуля та базову статистику прогресу. Це дозволило швидко протестувати бізнес-гіпотези та отримати фідбек від перших користувачів.

Після позитивних результатів MVP відбулася поступова інтеграція розширених модулів:

- Панель адміністратора для керування курсами;
- Фінансова аналітика;
- Механізми сертифікації та перевірки знань;
- Інтеграція з платіжними системами (LiqPay, Stripe).

Кожен новий функціональний блок проходив обов'язкову перевірку у staging-середовищі з подальшим деплоєм у production лише після валідації результатів QA-командою.

Нижче у фінальній версії роботи представлено скріншоти ключових модулів веб-сервісу для онлайн-курсів.


Рис. 4.2. демонструє інтерфейс авторизації користувача на веб-сервісі, рис. 4.3. – панель користувача з доступними курсами, рис. 4.4. – панель адміністратора, рис. 4.5. – Модуль проходження курсу.

## Login

Welcome back! Please log in to access your account.

**Email**

**Password** [Forgot Password?](#)




Remember Me

**Login**

---

OR

 Login with Google

Don't have an account? [Sign up →](#)

Рис. 4.2. Інтерфейс авторизації користувача

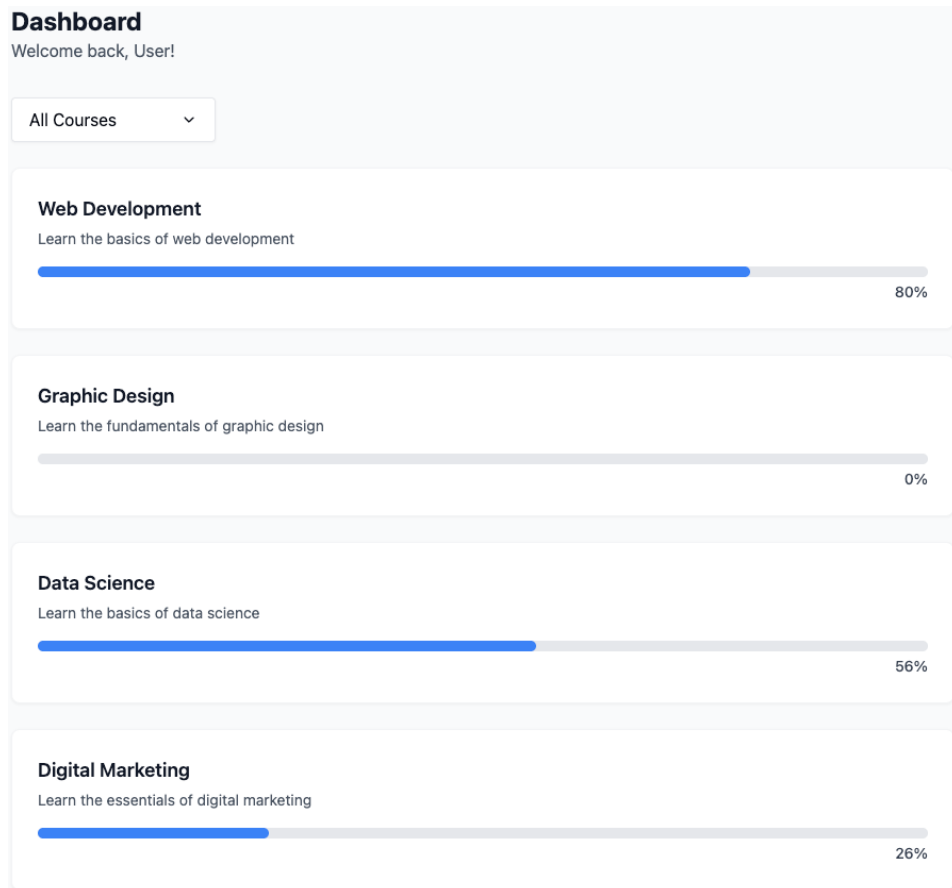


Рис. 4.3 Панель користувача з доступними курсами

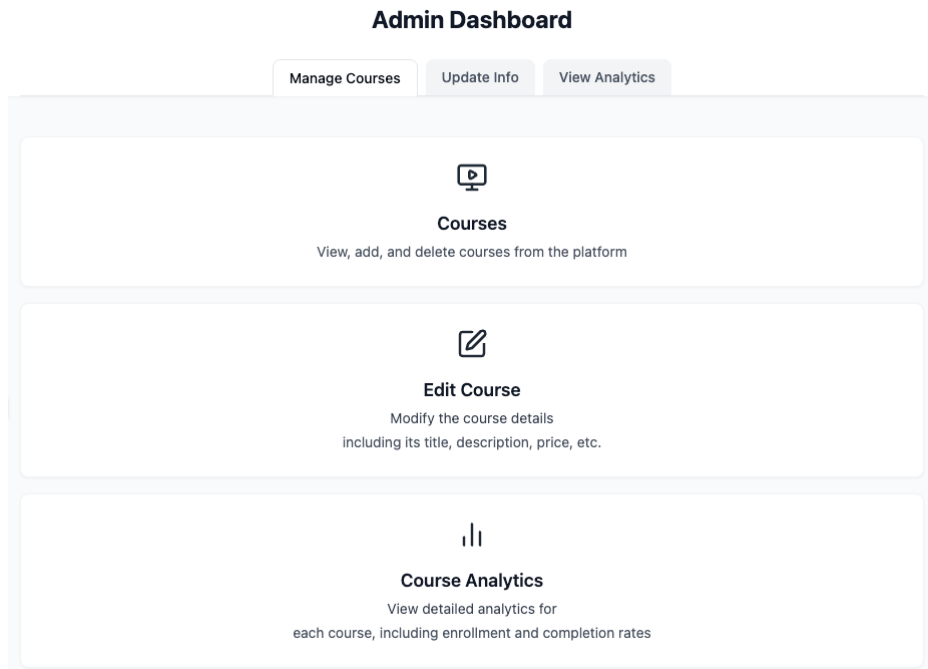


Рис. 4.4. Панель адміністратора

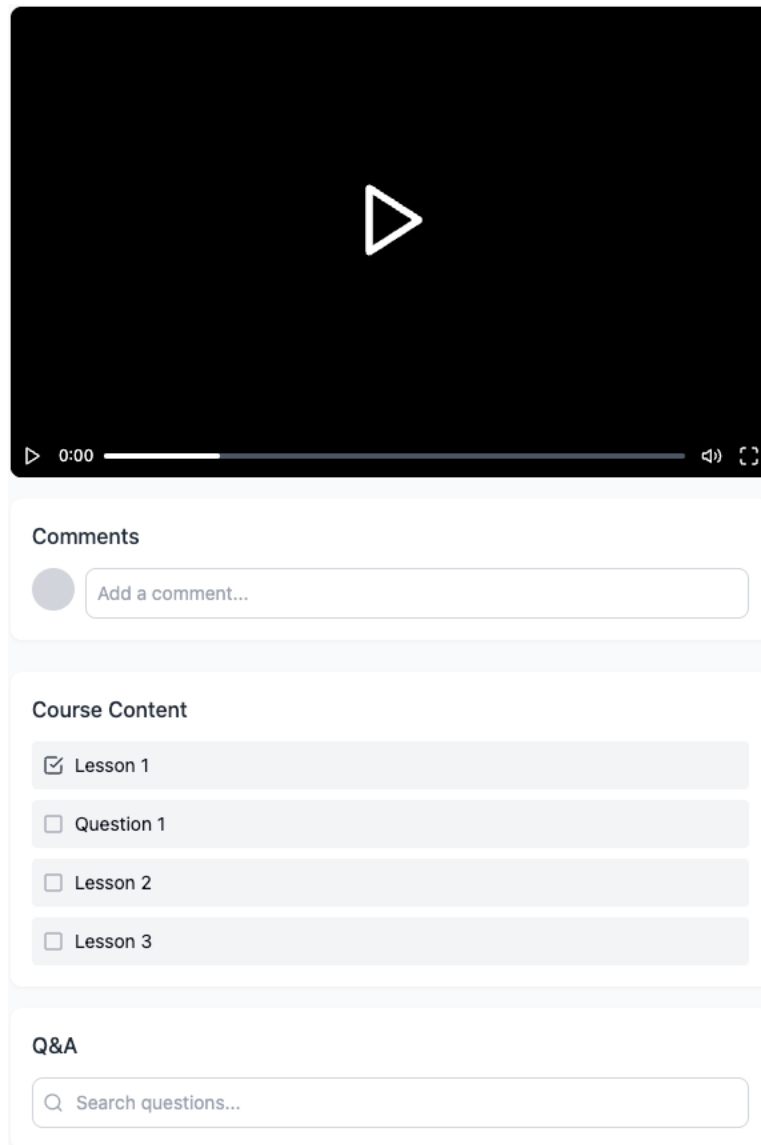


Рис. 4.5. Модуль проходження курсу

Також у роботі в додатках наведено код, реалізований в React, який ілюструє наступні функції:

1. Інтерфейс авторизації користувача (див. рис. А.1., А.2. додатка А);
2. Панель користувача з доступними курсами (див. рис. Б.1., Б.2. додатка Б);
3. Панель адміністратора (див. рис. В.1., В.2. додатка В);
4. Модуль проходження курсу (див. рис. Г.1., Г.2., Г.3. додатка Г).

Ще одним ключовим елементом стало впровадження автоматизованих тестів у пайплайн CI/CD:

- Unit-тести – покриття основних логічних модулів;
- Integration-тести – тестування зв'язків між сервісами (API ↔ DB);

- Smoke-тести – перевірка роботи головної сторінки, логіну, завантаження курсів;

- Manual QA – щотижневі тестування функціональності через staging. Результати логуються в CloudWatch, з автоматичним повідомленням у Slack у разі критичних збоїв.

Після впровадження системи до production середовища впроваджено механізми моніторингу й трекінгу:

- Google Analytics – відстеження поведінки користувачів;
- Sentry – сповіщення про frontend/backend помилки;
- Grafana – моніторинг навантаження на систему;
- AWS Auto Scaling – автоматичне масштабування інстансів у разі зростання трафіку.

Таким чином, впроваджена система готова до роботи з тисячами користувачів, включно з адаптацією під мобільні пристрої та подальшим розвитком функціональності.

#### **4.2 Імплементация функціональності та валідація рішень**

Етап імплементации функціональності є центральним для технічної реалізації веб-сервісу, оскільки саме на цьому етапі закладається програмна логіка, інтегруються окремі модулі та відбувається поступова перевірка відповідності реалізованих рішень початковим вимогам. З огляду на складну архітектуру системи, цей етап супроводжується постійною перевіркою коду, багаторівневим тестуванням, оновленням технічної документації та підготовкою до виведення продукту у production.

Функціональні модулі веб-сервісу реалізовувались у відповідності до попередньо затвердженої моделі архітектури. Вся логіка була поділена на кілька функціональних шарів:

- Presentation Layer – інтерфейс користувача (React), адаптивна верстка та компоненти UI/UX;
- Application Layer – API-рівень на Node.js/Express, який відповідає за обробку запитів;

- Business Logic Layer – модулі обробки курсів, аналітики, сертифікації;
- Data Layer – модулі взаємодії з базою даних PostgreSQL та кешем Redis.

Кожна частина реалізовувалась як окремий компонент із чітко визначеним інтерфейсом, що дозволяло незалежно тестувати та розгортати окремі частини без порушення цілісності проєкту.

Для забезпечення якості програмного продукту було впроваджено стратегію тестування, що охоплює три основні рівні:

1. Unit-тести – написані для кожного окремого модуля або функції, що не мають залежностей від інших компонентів. Покриття unit-тестами досягло 95% для API автентифікації, понад 80% – для курсової логіки.

2. Integration-тести – дозволили перевірити взаємодію між модулями API та базою даних, інтеграцію з поштовими сервісами та платіжними шлюзами. Більшість критичних точок, таких як обробка платежів, отримали окремі тести з емуляцією реального сценарію.

3. Manual QA – команда тестувальників перевіряла поведінку інтерфейсів у staging-середовищі. Були протестовані сценарії реєстрації, проходження курсу, адміністрування матеріалів, згенеровані чек-листи.

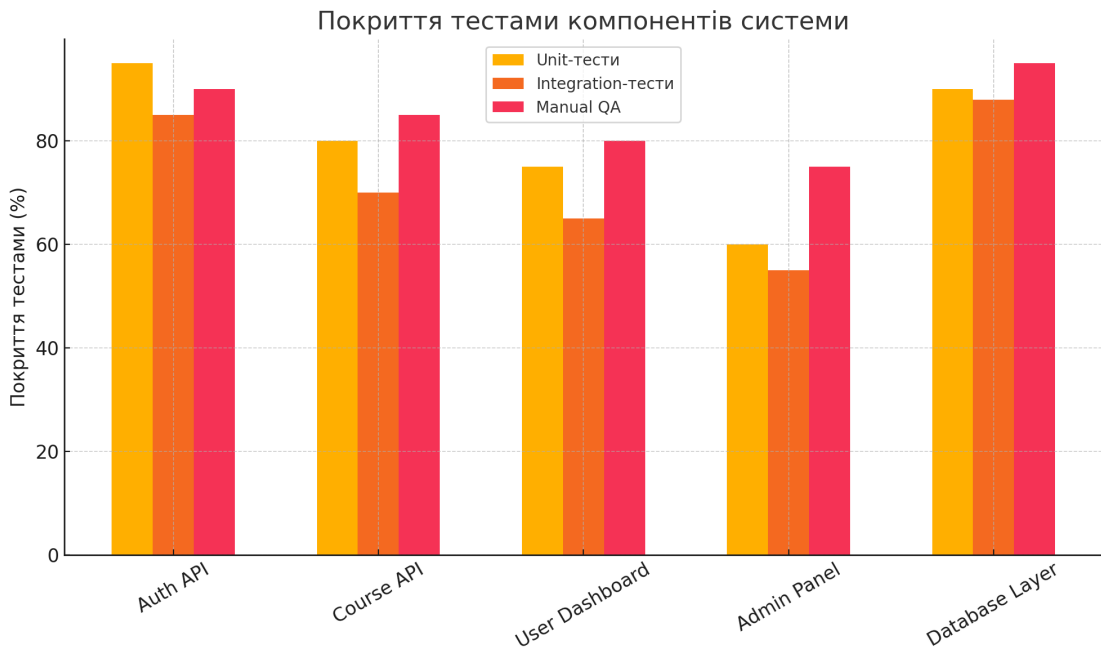


Рис. 4.6. Покриття тестами компонентів системи

На діаграмі (рис. 4.6.) відображено співвідношення трьох рівнів тестування (unit, integration, manual) для основних модулів: API автентифікації, API курсів, панелі користувача, адмін-панелі та взаємодії з базою даних. Найбільше покриття спостерігається в критичних для безпеки та доступності зонах – автентифікації та базі даних.

У процесі реалізації проєкту були виявлені та вчасно усунуті кілька критичних архітектурних обмежень:

- Проблема надмірної зв'язаності компонентів – вирішено через впровадження REST API gateway;
- Вузьке місце в логіці оновлення матеріалів – оптимізовано кешуванням на рівні CDN;
- Недостатня продуктивність фронтенду на мобільних пристроях – оновлено UI-компоненти для адаптивної поведінки.

Всі ці коригування повинні бути зафіксовані в технічній документації та погоджені з розробниками та аналітиками.

Технічна документація оновлювалась паралельно з розробкою функціональності та містить такі блоки:

- API-документація (Swagger) – перелік ендпоінтів, параметри, відповіді, статус-коди;
- Архітектурна документація – опис інтерфейсів між сервісами, структура модулів;
- DevOps інструкції – алгоритм розгортання, деплоймент-скрипти, змінні середовища;
- Тест-кейси та сценарії – список перевірок із результатами та помилками.

Усі документи зберігаються в централізованому сховищі Notion та GitLab Wiki з відповідними правами доступу.

Завдяки системному підходу до тестування та архітектурної гнучкості веб-сервіс забезпечує відповідність розробки заявленим функціональним вимогам, стабільність роботи навіть при збільшенні навантаження, легкість масштабування, контрольованість оновлень та розширень.

Проведений аналіз доводить, що впровадження функціональності сервісу здійснено відповідно до сучасних стандартів ІТ-проєктування: високий рівень покриття тестами, структурований супровід документації та оперативне усунення архітектурних недоліків. Це створює технічну основу для стабільної роботи сервісу та його подальшого масштабування.

#### **4.3 Система контролю прогресу та управління відхиленнями**

Контроль за виконанням плану проєкту та оперативне управління відхиленнями є однією з найважливіших складових ефективного управління ІТ-проєктом. Для реалізації веб-сервісу онлайн-курсів, що базується на складній архітектурі така система набуває особливого значення. Вона забезпечує постійне відстеження виконання завдань, своєчасну ідентифікацію відхилень, ефективну комунікацію між членами команди та динамічне оновлення планів у відповідь на зовнішні чи внутрішні зміни.

Для контролю реалізації проєкту було сформовано деталізований графік виконання робіт, представлений у форматі діаграми Ганта. Цей інструмент дозволяє не лише візуалізувати часову логіку етапів, а й ідентифікувати критичний шлях, зони перетину відповідальностей і можливі точки ризику.

Основні етапи, які були включені до графіку:

- Ініціалізація проєкту (01.10.2024 – 10.10.2024)
- Аналіз вимог і створення технічної документації (10.10.2024 – 20.10.2024)
- Дизайн інтерфейсу користувача (21.10.2024 – 03.11.2024)
- Розробка backend-модуля (04.11.2024 – 25.11.2024)
- Розробка frontend-модуля (04.11.2024 – 25.11.2024)
- Комплексне тестування (04.12.2024 – 20.12.2024)
- Впровадження та публічний запуск (10.01.2025 – 21.01.2025)

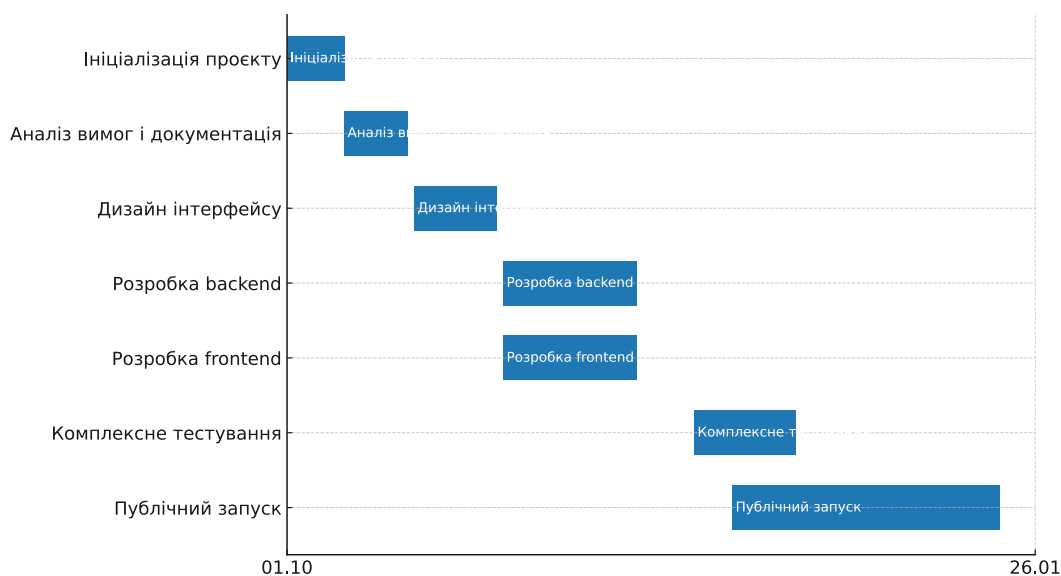


Рис. 4.7. Діаграма Ганта для контролю прогресу

Для щоденного контролю виконання задач команда використовувала Jira як основну систему трекінгу задач, із розбиттям за етапами спринтів, Google Calendar та Slack для нагадувань про щоденні стендапи, планування ретроспектив. Кожен член команди має доступ до планів, що дозволяє прозоро бачити статус проєкту на будь-якому етапі.

Беклог проєкту був побудований відповідно до Agile-методології з поділом задач на пріоритетні, другорядні та технічні. Для кожного спринту

проводилося попереднє планування (Sprint Planning), оцінка обсягу задач (story points), декомпозиція задач до рівня технічного завдання. Менеджер проєкту щоденно оновлював статус беклогу згідно зі звітами команди. У випадку виявлення затримок чи блокерів команда використовувала систему внутрішніх «алертів» для оперативної реакції (Ping в Slack, призначення відповідальних, термінові обговорення).

Під час реалізації проєкту ідентифіковано низку ризиків, які могли спричинити відхилення від графіку (таблиця 4.1.).

*Таблиця 4.1.*

**Таблиця ризиків відхилення від графіку**

<b>Ідентифікований ризик</b>	<b>Потенційний вплив</b>	<b>Запроваджена контрдія</b>
Втрата розробника на критичному етапі	Затримка backend-розробки	Створено резервний пул dev-ресурсів
Технічна нестабільність API	Неможливість інтеграції з frontend	Запроваджено окреме середовище для тестів
Відсутність фідбеку від користувачів	Неможливість уточнення UX-рішень	Впроваджено швидкі опитування та аналіз логів
Проблеми з хостингом staging-серверу	Втрати часу на деплой і QA	Міграція на стабільне AWS EC2 середовище

У разі виникнення відхилень будуть проводитись ревізії діаграми Ганта з урахуванням нових даних, переглядись послідовності задач (реорганізація беклогу), залучатися додаткові ресурси, скорочуватись non-critical задачі для збереження дедлайну.

#### **4.4 Управління командною взаємодією та цифровими інструментами**

Управління командною взаємодією є одним з ключових факторів успішного впровадження будь-якого ІТ-проєкту, особливо у випадках, коли проєкт реалізується в умовах розподіленої команди або високої інтенсивності

спринтів. Цей аспект стає критичним у проєктах розробки онлайн-сервісів для навчання, де необхідно забезпечити синхронність між технічною, аналітичною та дизайнерською частинами розробки, а також дотримання термінів, ефективний розподіл задач і швидку реакцію на відхилення.

Щоденні короткі мітинги – так звані daily standups – відіграють фундаментальну роль в синхронізації роботи учасників проєкту. Вони дозволяють відстежити поточний прогрес кожного члена команди, виявити блокери на ранніх етапах, забезпечити прозору комунікацію між віддаленими учасниками. Стендапи проводилися щоранку о 10:00 у форматі 15-хвилинного дзвінка в Zoom. Кожен учасник дає коротку відповідь на три питання: що зроблено за минулий день, що планується сьогодні, і чи є якісь труднощі. Менеджер проєкту фіксує проблеми, які потребують додаткової уваги у системі Jira.

Jira використовується для створення технічних завдань, контролю спринтів, звітності про виконання задач. Кожен спринт має власну дошку з пріоритизацією за допомогою story points. Також були налаштовані шаблони задач: Epic, Story, Task, Subtask.

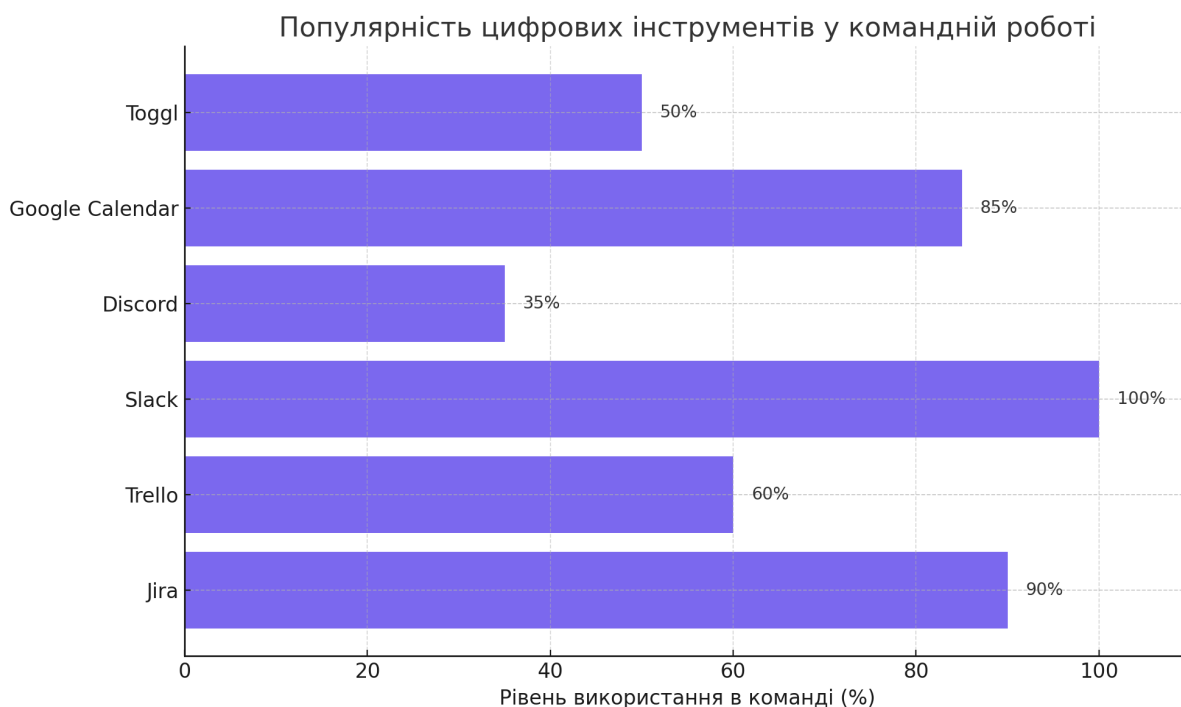


Рис. 4.8. Популярність цифрових інструментів у командній роботі

На рисунку представлено порівняльний рівень використання основних цифрових інструментів командою. Найбільшу популярність має Slack як основна платформа для внутрішньої комунікації, далі йде Jira, яка є ядром для керування задачами. Google Calendar використовується для планування дзвінків та дедлайнів, а Trello, Discord і Toggl для допоміжних завдань.

Основним каналом щоденного спілкування в команді є Slack. Були створені окремі канали для кожної функціональної зони: #frontend, #backend, #design, #qa, #analytics, #support, а також канали #daily, #urgent, #release. Slack дозволив миттєво ділитися файлами, кодом, повідомленнями, створювати внутрішні опитування, а також інтегрувати повідомлення зі сторінки деплоїв, CI/CD-пайплайнів тощо.

Discord використовувався переважно для неформального спілкування або довгих асинхронних обговорень. Такий поділ дозволив зменшити “шум” у Slack і підтримати здорову атмосферу всередині колективу.

Google Calendar є базовим інструментом для синхронізації зустрічей, термінів спринтів, стендапів, демо та ретроспектив. Кожен член команди має доступ до загального календаря проєкту, що дозволяє уникати конфліктів у розкладі та забезпечувало повну прозорість.

Окрім щоденних стендапів, у розкладі команди є щотижнева синхронізація (понеділок, 45 хвилин), обговорення досягнень за попередній тиждень, розподіл задач на поточний спринт, ретроспектива (кожні два тижні, 60 хвилин), визначення сильних/слабких сторін, аналіз помилок, демо-дзвінки (після кожного великого оновлення), презентація досягнень перед стейкхолдерами, peer-review-сесії (раз на тиждень), код-рев'ю між розробниками, обговорення складних кейсів. Ці регулярні зустрічі не лише сприяють дисципліні, а й створюють основу для рефлексії, командного навчання та удосконалення внутрішніх процесів.

Особлива увага приділяється темам психологічного вигорання, work-life balance та рівномірного навантаження. Менеджер щотижня проводив mini-1-on-1 дзвінки для моніторингу морального стану команди.

У результаті впровадження цілісної системи цифрової взаємодії досягнуто високого рівня прозорості в роботі команди, зменшено ризики комунікаційних збоїв, забезпечено синхронність між різними підгрупами команди (розробка, тестування, дизайн), суттєво підвищено мотивацію та залученість учасників.

#### **4.5 Управління стейкхолдерами та стратегічна комунікація**

Управління стейкхолдерами є невіддільною частиною проєктного менеджменту в сфері ІТ, де значна кількість зацікавлених сторін впливає на всі фази життєвого циклу проєкту. У випадку розробки веб-сервісу для онлайн-курсів у форматі EdTech, коло стейкхолдерів охоплює не лише внутрішніх учасників (розробники, дизайнери, тестувальники, проєктний менеджмент), але й зовнішніх (інвестори, кінцеві користувачі, освітні партнери, маркетологи, партнери з дистрибуції). Управління взаємодією з ними має бути системним, адаптивним та відповідати принципам відкритої прозорості комунікації.

На першому етапі було здійснено повну ідентифікацію всіх груп стейкхолдерів, із зазначенням їхніх очікувань, ролей, каналів впливу та можливого рівня зацікавленості. Класифікацію здійснено за такими категоріями:

- Внутрішні стейкхолдери: проєктна команда (PM, розробники, дизайнери, QA, бізнес-аналітики), менеджери ресурсів, технічні консультанти.
- Зовнішні стейкхолдери: замовник (інвестор або підприємство), кінцеві користувачі (учні та викладачі), контрагенти (хостинг-платформи, постачальники API), представники державних органів (у частині дотримання законодавства про захист даних).

Кожен стейкхолдер отримав оцінку за критеріями впливу на проєкт і зацікавленості у його результатах. Результати відображено у вигляді мапи (рис. 4.9.).

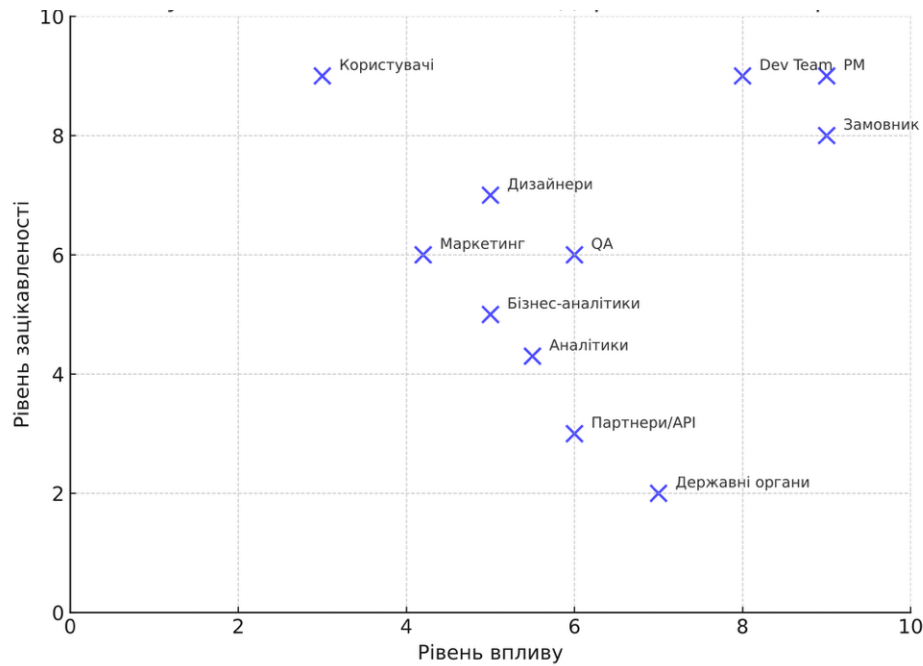


Рис. 4.9. Мапа стейкхолдерів: вплив/інтерес

На представленій діаграмі чітко видно позиціонування основних гравців. Найвищий рівень впливу та інтересу мають РМ, замовник та Dev Team, оскільки саме вони приймають стратегічні рішення та реалізують продукт. Користувачі мають високий інтерес, але обмежений вплив – проте їхнє задоволення прямо впливає на успіх проєкту. Маркетинговий відділ та аналітики мають середній вплив, оскільки відповідають за ринкову адаптацію сервісу.

Після класифікації кожної групи визначено оптимальні підходи до комунікації. У табличному вигляді узагальнено типові стратегії:

Таблиця 4.2.

**Стратегії взаємодії зі стейкхолдерами за рівнем впливу та інтересу**

Категорія	Стейкхолдери	Стратегія взаємодії
Високий вплив / Високий інтерес	РМ, замовник	Постійне залучення, щотижневі зустрічі, повний доступ до звітів і ходу робіт
Високий вплив / Низький інтерес	Державні органи	Взаємодія в межах юридичних вимог, підтримка репутації

Продовження табл. 4.2.

Низький вплив / Високий інтерес	Користувачі	Аналітика поведінки, UX-опитування, бета- тестування
Низький вплив / Низький інтерес	Вторинні постачальники	Базовий інформаційний зв'язок, реагування за потреби

Визначено канали взаємодії залежно від стейкхолдера:

Таблиця 4.3.

### Канали взаємодії з стейкхолдерами

Категорія стейкхолдера	Канал комунікації	Частота	Формат
Замовник / Інвестор	Email, Zoom, щомісячний звіт	Щотижнево	Статус-апдейти, KPI, бюджет
Команда розробки	Slack, Jira, стендапи	Щоденно	Таски, оновлення, блокери
Кінцеві користувачі	UX-опитування, аналітика	Раз на спринт	Анкети, інтерв'ю, heatmaps
Маркетинг	Trello, Notion, Email	Двічі на тиждень	Планування кампаній
QA та підтримка	Slack, TestRail, Google Docs	Щоденно	Результати тестів, оновлення

Ключовим у цій матриці є диференціація за форматом: для замовника важливі звіти, для команди – беклог і актуальні таски, для користувачів – UX-підтримка.

На етапі початкової аналітики було проведено глибинні інтерв'ю з основними групами, де замовники окреслили очікування від MVP, допустимий

бюджет і часові рамки, РМ та техліди визначили технічні обмеження та залежності, UX-фахівці провели юзабіліті-інтерв'ю з тестовими користувачами. Це дозволило налаштувати правильні очікування, усунути технічні бар'єри ще до початку розробки та знизити ризики на пізніших етапах.

Крім мапи та стратегії, створено матрицю RACI, яка дозволяє чітко розподілити відповідальність:

*Таблиця 4.4.*

### Матриця RACI

Завдання	Відповідальний (R)	Узгоджує (A)	Консультує (C)	Поінформований (I)
Розробка техдизайну	UI Designer	PM	Dev Team	Stakeholders
Вибір технологічного стеку	СТО	PM	Dev Team	Замовник
UX- дослідження	Аналітик	PM	Користувачі	Команда
Розробка MVP	Dev Team	PM	QA, СТО	Замовник

У межах взаємодії з найвпливовішими стейкхолдерами було запроваджено механізм “crisis escalation path”, який передбачає, виявлення конфліктної ситуації або потенційного ризику (затримка, недосягнення KPI).

Системне управління стейкхолдерами дозволило виявити потенційні ризики, краще зрозуміти очікування зацікавлених сторін та оптимізувати інформаційні потоки. Ідентифікація груп за критеріями впливу та інтересу, побудова мапи та стратегічної матриці комунікацій і спільне планування дозволили підвищити залученість, забезпечити стабільність та впровадити інноваційні рішення. У наступних фазах така модель комунікації продовжить функціонувати як основа масштабування та підтримки якості розвитку платформи.

## ВИСНОВКИ

Проведене дослідження процесів управління проектом розробки веб-сервісу для онлайн-курсів дозволило розглянути теоретичні, методологічні та практичні аспекти створення складного інформаційного продукту в умовах динамічного цифрового середовища. Робота поєднує в собі принципи класичного управління проектами з адаптивними методологіями, сучасні інструменти DevOps та моделі аналізу продуктивності IT-систем.

Було обґрунтовано актуальність розвитку онлайн-освіти як стійкого глобального тренду, підтвердженого аналітичними даними. Серед основних переваг таких платформ виокремлено доступність, масштабованість, інклюзивність, а також можливість персоналізації навчання. Встановлено, що управління IT-проектами у сфері EdTech вимагає особливої уваги до UX-дизайну, якості навчального контенту, гнучкості архітектури системи.

Проведено концептуалізацію продукту: визначено цілі, цільову аудиторію, функціональні блоки та структуру взаємодії користувачів. Здійснено моделювання бази даних, формалізовано логіку роботи платформи через архітектурну модель, що поєднує frontend, backend, систему тестування, сертифікації, аналітики та персоналізації. Архітектура сервісу розроблена з урахуванням принципів масштабованості, безпеки та можливості контейнеризації з використанням Docker. Інтеграція хмарної інфраструктури (AWS), моніторингових інструментів та автоматизованого CI/CD циклу створює підґрунтя для безперервного розгортання та оновлення системи без зупинки сервісу.

Досліджено управлінські аспекти реалізації проекту. Здійснено декомпозицію цілей у вигляді дерева, розроблено WBS за фазами життєвого циклу та за функціональними модулями, що дозволило поєднати класичне і гнучке планування. Визначено ролі членів команди, побудовано організаційну структуру, описано SCRUM-процеси. Застосовано діаграму Ганта для побудови календарного плану, проаналізовано критичний шлях, визначено контрольні точки та буфери часу. Проведено SWOT-аналіз, який виявив основні можливості,

ризиками та внутрішні обмеження проєкту. Запропоновано стратегії їх подолання, включаючи багаторівневе тестування, партнерську модель залучення контенту та поступове масштабування платформи. Оцінено фінансову модель на стартовій фазі, побудовано прогноз cash-flow на 8 місяців після запуску, визначено точку безбитковості. Застосовано модель PSI для оцінки вірогідності успіху, враховуючи чинники поведінки команди, чіткість вимог, залученість користувачів і підтримку керівництва.

Деталізовано технічну інфраструктуру: розподіл середовищ (development, staging, production), реалізацію пайплайнів CI/CD на базі GitHub Actions, розгортання через Nginx, контейнеризацію та балансування навантаження. Проведено моделювання серверного навантаження, що дало змогу визначити граничні навантаження та обґрунтувати масштабування інфраструктури. Аналіз ризиків проведено в розрізі технічних, організаційних і ринкових факторів. Складено карту зацікавлених сторін (stakeholder map) з їх класифікацією за рівнем впливу та інтересу, визначено відповідні комунікаційні стратегії. Розроблено RACI-матрицю, яка чітко структурує відповідальність команди в ключових процесах. Запропоновано канали зв'язку, частоту взаємодії та типи звітності для кожної групи стейкхолдерів.

На підставі всіх результатів можна зробити висновок, що ефективно управління проєктом у сфері онлайн-освіти можливе лише за умови інтеграції інженерного та управлінського підходів. Робота демонструє, що створення якісного EdTech-продукту потребує не лише програмного забезпечення, але й побудови цілісної екосистеми з урахуванням користувацького досвіду та економічної життєздатності. Запропонована модель може слугувати основою для створення подібних сервісів у суміжних сферах – від професійного навчання до корпоративних LMS. Застосування системної аналітики, математичних моделей і розрахунків обґрунтовує технічні рішення і підвищує якість управлінських дій.

## ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Technology in education - 2023 GEM Report. 2023 GEM Report. URL: <https://gem-report-2023.unesco.org/technology-in-education/#:~:text=During%20school%20closures,%2095,some%20countries%20building%20on%20decades>
2. What you need to know about digital learning and transformation of education. UNESCO, 17 September 2024. URL: <https://www.unesco.org/en/digital-education/need-know#:~:text=conflicts.%20During%20the%20COVID,the%20relevance%20and%20quality%20of>
3. OECD Digital Education Outlook 2023. Towards an Effective Digital Education Ecosystem. 13 December 2023. URL: [https://www.oecd.org/en/publications/2023/12/oecd-digital-education-outlook-2023\\_c827b81a.html#:~:text=Almost%20half%20of%20the%20countries,19%20pandemic](https://www.oecd.org/en/publications/2023/12/oecd-digital-education-outlook-2023_c827b81a.html#:~:text=Almost%20half%20of%20the%20countries,19%20pandemic)
4. Sizing the Global EdTech Market. Mode vs Model. HolonIQ. Global Impact Intelligence. URL: <https://www.holoniq.com/notes/sizing-the-global-edtech-market#:~:text=EdTech%20is%20growing%20at%2016.3,global%20education%20market%20by%202025>
5. Ukraine's recovery is dependent on the recovery of children's education. Statement by UNICEF Regional Director for Europe and Central Asia, Regina De Dominicis. 12 June 2024. URL: <https://www.unicef.org/press-releases/ukraines-recovery-dependent-recovery-childrens-education#:~:text=%E2%80%9CBecause%20of%20COVID,person%20school%20at%20all>
6. Nwaka U. EdX vs. Coursera vs. Udemy: Deciding Where to Upskill for Professionals. Medium. URL: <https://medium.com/@nwakauc1/edx-vs-coursera-vs-udemy-deciding-where-to-upskill-for-professionals-35f6f8fb6aba>

7. Coursera vs Udemy 2025: Ultimate Comparison & Review. Brian Vander Waal. URL: <https://briandvanderwaal.com/coursera-vs-udemy#:~:text=,audit%20many%20courses%20without%20paying>
8. The Effectiveness of Gamification in Online Learning | Journal of Online and Distance Learning. International Peer Reviewed Journals Database - CARI Journals. URL: [https://carijournals.org/journals/index.php/JODL/article/view/1693?srsltid=AfmBOoqjpryUyMN2J\\_Vq-8C\\_\\_h4QFYulJefNrv9VIGJM0H1Yl2xiM6Ud#:~:text=Journals%20carijournals,engagement,%20motivation,%20and%20learning%20outcomes](https://carijournals.org/journals/index.php/JODL/article/view/1693?srsltid=AfmBOoqjpryUyMN2J_Vq-8C__h4QFYulJefNrv9VIGJM0H1Yl2xiM6Ud#:~:text=Journals%20carijournals,engagement,%20motivation,%20and%20learning%20outcomes)
9. By The Numbers: MOOCs in 2021 – Class Central. The Report by Class Central. URL: <https://www.classcentral.com/report/mooc-stats-2021/#:~:text=Now,%20a%20decade%20later,%20MOOCs,3100%20courses%20and%20500%20microcredentials>
10. 50 Online Education Statistics: 2025 Data on Higher Learning & Corporate Training. Imed Bouchrika, April 9, 2025. URL: <https://research.com/education/online-education-statistics#:~:text=1,VR%29%2C%20and%20augmented>
11. The PMBOK's Project Management Documents. ProjectEngineer. URL: <https://www.projectengineer.net/the-pmboks-project-management-documents/#:~:text=,stakeholders%20around%20a%20common%20vision>
12. Work Breakdown Structure (WBS) Joe Tinger. Jul 2, 2019. URL: [https://www.projectmanagement.com/wikis/397395/work-breakdown-structure--wbs-#\\_](https://www.projectmanagement.com/wikis/397395/work-breakdown-structure--wbs-#_)
13. Морозов В.В. Управління проектами розвитку підприємств : [навч. посіб.] / В. В. Морозов, О. В. Кальніченко, Ю. Г. Турло ; Ун-т економіки та права «КРОК». – К. : ВНЗ "Ун-т економіки та права «КРОК», 2011. – 231 с.
14. Морозов В. В. Модель впливу зовнішнього оточення на процес управління конфігурацією в проекті / В. В. Морозов, С. И. Рудницький // Управління розвитком складних систем. - 2013. - Вип. 16. - С. 40-52.

15. В.В. Морозов, О.Г. Тімінський, А.С. Коломієць. Методичні вказівки до написання кваліфікаційної роботи магістра. Київ – 2024
16. Морозов В.В. Чередніченко А.М., Шпильова Т.І. «Формування, управління та розвиток команди проекту». Київ, 2009. Бушуєв С. Д. Методології управління проектами на моделях класу "рушійні сили – опори" / С. Д. Бушуєв, Р. Ф. Ярошенко // Управління розвитком складних систем. - 2010. - Вип. 2. - С. 11-14.
17. Scott Berkun - Making Things Happen: Mastering Project Management (Theory in Practice). 2008.
18. Project Management Institute - A Guide to the Project Management Body of Knowledge (PMBOK Guide) – and the Standard for Project Management. 2021.
19. Jeff Sutherland - Scrum: the art of doing twice the work in half the time. 2014.
20. Ken Schwaber - Agile Project Management With Scrum. 2004.
21. Moh'd A. Radaideh - Software Project Management: With PMI, IEEE-CS and Agile-SCRUM (de Gruyter Textbook). 2023.
22. Walker, Christopher R. - Minimalist Money: The Lean Startup's Path to Financial Freedom. 2024.
23. Eric Verzuh - The Fast Forward MBA in Project Management. 2021.
24. Bokish Ltd; Daniel H. Pink - Drive: The Surprising Truth About What Motivates Us. 2015.
25. Kenneth S. Rubin - Essential Scrum: A Practical Guide to the Most Popular Agile Process. 2015.
26. AXELOS, The Stationery Office (editor) - Managing Successful Projects with PRINCE2 2017 Edition. 2018.
27. Agile-Concurrent hybrid: A framework for concurrent product development using Scrum / T. Žužek et al. Concurrent Engineering. 2020. Vol. 28, no. 4. P. 255–264.
28. Langerman J., Leung W. S. The effect of outsourcing and insourcing on Agile and DevOps. Journal of Information Technology Teaching Cases. 2023. P. 204388692311768.

29. Agile-Concurrent hybrid: A framework for concurrent product development using Scrum / T. Žužek et al. *Concurrent Engineering*. 2020. Vol. 28, no. 4. P. 255–264.
30. Mishra M. Covid-19 and school education in India: A critical spatial and ‘meritocratic’ appraisal of emerging EdTech. *Power and Education*. 2023.
31. Fakih A. Analysing the Contributions of Edtech Startups to the Corporate Sector. *Journal of Contemporary Business Research*. 2025.
32. Kim (김덕순) D., Borowiec K., Wortham S. A New Era in EdTech: Emerging Challenges and Opportunities. *ECNU Review of Education*. 2023.
33. O. Stetsyk, P. Pasiaka and B. Yeremenko, “Designing an Effective System Architecture for Detecting Propaganda and Spam in Social Media News Feed,” 2023 IEEE 4th KhPI Week on Advanced Technology (KhPIWeek), Kharkiv, Ukraine, 2023.
34. Poliakov M., Yeremenko B. Info-Communication System of the Electronic Estimation of Adolescents’ Special Abilities. *Management of Development of Complex Systems*. 2024. No. 58. P. 139–145. URL: <https://doi.org/10.32347/2412-9933.2024.58.139-145>.

## ДОДАТОК А

### Код, реалізований в React, який ілюструє інтерфейс авторизації користувача

```
1 import { useState, useContext } from 'react';
2 import { Eye, EyeOff } from 'lucide-react';
3 import { AuthServiceContext } from './AuthServiceContext';
4
5 export default function LoginPage() {
6   const authService = useContext(AuthServiceContext);
7   const [email, setEmail] = useState("");
8   const [password, setPassword] = useState("");
9   const [showPassword, setShowPassword] = useState(false);
10  const [rememberMe, setRememberMe] = useState(false);
11
12  const handleSubmit = async (e) => {
13    e.preventDefault();
14    const result = await authService.login(email, password);
15    console.log("Login result:", result);
16  };
17
18  const handleGoogleLogin = async () => {
19    const result = await authService.loginWithGoogle();
20    console.log("Google login result:", result);
21  };
22
23  return (
24    <div className="bg-gray-100 min-h-screen flex items-center justify-center p-4">
25      <div className="bg-white rounded-lg shadow-md p-8 w-full max-w-md">
26        <div className="text-center mb-6">
27          <h1 className="text-2xl font-semibold">Login</h1>
28          <p className="text-gray-500 text-sm mt-1">Welcome back! Please log in to access your account.</p>
29        </div>
30        <form onSubmit={handleSubmit} className="space-y-4">
31          <div>
32            <label className="block text-sm font-medium text-gray-700">Email</label>
33            <input
34              type="email"
35              value={email}
36              onChange={(e) => setEmail(e.target.value)}
37              className="mt-1 block w-full px-3 py-2 border border-gray-300 rounded-md shadow-sm focus:outline-none focus:ring focus:ring-indigo-200"
38              required
39            />

```

Рис. А.1.

```

40 </div>
41 <div>
42   <label className="block text-sm font-medium text-gray-700">Password</label>
43   <div className="relative">
44     <input
45       type={showPassword ? 'text' : 'password'}
46       value={password}
47       onChange={(e) => setPassword(e.target.value)}
48       className="mt-1 block w-full px-3 py-2 border border-gray-300 rounded-md shadow-sm focus:outline-none focus:ring focus:ring-indigo-200"
49       required
50     />
51     <button
52       type="button"
53       onClick={() => setShowPassword(!showPassword)}
54       className="absolute inset-y-0 right-0 pr-3 flex items-center text-gray-500"
55     >
56       {showPassword ? <EyeOff size={18} /> : <Eye size={18} />}
57     </button>
58   </div>
59 </div>
60 <div className="flex items-center">
61   <input
62     type="checkbox"
63     checked={rememberMe}
64     onChange={(e) => setRememberMe(e.target.checked)}
65     className="h-4 w-4 text-indigo-600 border-gray-300 rounded"
66   />
67   <label className="ml-2 block text-sm text-gray-900">Remember me</label>
68 </div>
69 <button
70   type="submit"
71   className="w-full py-2 px-4 bg-indigo-600 text-white rounded hover:bg-indigo-700"
72 >
73   Login
74 </button>
75 </form>
76 <button
77   onClick={handleGoogleLogin}
78   className="mt-4 w-full py-2 px-4 bg-red-500 text-white rounded hover:bg-red-600"
79 >
80   Login with Google
81 </button>
82 </div>
83 </div>
84 )

```

Рис. А.2.

## ДОДАТОК Б

Код, реалізований в React, який ілюструє панель користувача з доступними курсами

```
1 import { useState, useEffect } from 'react';
2 import { ChevronDown } from 'lucide-react';
3
4 export default function Dashboard() {
5   const [selectedFilter, setSelectedFilter] = useState('All Courses');
6   const [showFilterDropdown, setShowFilterDropdown] = useState(false);
7   const [courses, setCourses] = useState([]);
8   const [loading, setLoading] = useState(true);
9   const [error, setError] = useState(null);
10
11   useEffect(() => {
12     const fetchCourses = async () => {
13       try {
14         const response = await fetch('https://api.denaksoy.com/courses');
15         if (!response.ok) throw new Error('Failed to fetch courses');
16         const data = await response.json();
17         setCourses(data);
18       } catch (err) {
19         setError(err.message);
20       } finally {
21         setLoading(false);
22       }
23     };
24
25     fetchCourses();
26   }, []);
27
28   const filterOptions = ['All Courses', 'In Progress', 'Completed', 'Not Started'];
29
30   const handleFilterSelect = (option) => {
31     setSelectedFilter(option);
32     setShowFilterDropdown(false);
33   };
34
35   const filteredCourses = courses.filter((course) => {
36     if (selectedFilter === 'All Courses') return true;
37     if (selectedFilter === 'In Progress') return course.progress > 0 && course.progress < 100;
38     if (selectedFilter === 'Completed') return course.progress === 100;
39     if (selectedFilter === 'Not Started') return course.progress === 0;
40     return true;
41   });
42
```

Рис. Б.1.

```

44 <div className="p-6">
45 <div className="flex justify-between items-center mb-4">
46 <h1 className="text-2xl font-bold">Dashboard</h1>
47 <div className="relative inline-block text-left">
48 <button
49   onClick={() => setShowFilterDropdown(showFilterDropdown)}
50   className="inline-flex justify-center items-center px-4 py-2 border border-gray-300 shadow-sm text-sm font-medium rounded-md bg-white hover:bg-gray-50"
51 >
52   {selectedFilter}
53 <ChevronDown className="ml-2 h-4 w-4" />
54 </button>
55 {showFilterDropdown && (
56 <div className="origin-top-right absolute right-0 mt-2 w-56 rounded-md shadow-lg bg-white ring-1 ring-black ring-opacity-5">
57 <div className="py-1">
58   {filterOptions.map((option) => (
59 <button
60   key={option}
61   onClick={() => handleFilterSelect(option)}
62   className="block w-full text-left px-4 py-2 text-sm text-gray-700 hover:bg-gray-100"
63 >
64     {option}
65 </button>
66   ))}
67 </div>
68 </div>
69 )}
70 </div>
71 </div>
72
73 {loading && <p>Loading courses...</p>}
74 {error && <p className="text-red-500">Error: {error}</p>}
75
76 {!loading && !error && (
77 <div className="grid grid-cols-1 md:grid-cols-2 gap-4">
78   {filteredCourses.map((course) => (
79 <div key={course.id} className="border rounded-lg p-4 shadow-sm">
80 <h2 className="text-lg font-semibold">{course.title}</h2>
81 <p className="text-sm text-gray-600 mb-2">{course.description}</p>
82 <div className="w-full bg-gray-200 rounded-full h-2.5">
83 <div
84   className="bg-indigo-600 h-2.5 rounded-full"
85   style={{ width: ${course.progress}% }}
86 ></div>
87 </div>
88 <p className="text-sm text-gray-500 mt-1">{course.progress}% complete</p>
89 </div>
90   ))}

```

Рис. Б.2.

## ДОДАТОК В

### Код, реалізований в React, який ілюструє панель адміністратора

```
1 import { useState } from 'react';
2 import { MonitorPlay, PenSquare, BarChart2 } from 'lucide-react';
3
4 export default function AdminDashboard() {
5   const [activeTab, setActiveTab] = useState('Manage Courses');
6
7   const tabs = [
8     'Manage Courses',
9     'Update Info',
10    'View Analytics'
11  ];
12
13  const actionCards = [
14    {
15      id: 1,
16      title: 'Courses',
17      description: 'View, add, and delete courses from the platform',
18      icon: <MonitorPlay className="w-8 h-8 text-gray-800" />,
19    },
20    {
21      id: 2,
22      title: 'Edit Course',
23      subtitle: 'Modify the course details',
24      description: 'including its title, description, price, etc.',
25      icon: <PenSquare className="w-8 h-8 text-gray-800" />,
26    },
27    {
28      id: 3,
29      title: 'Course Analytics',
30      subtitle: 'View detailed analytics for',
31      description: 'each course, including enrollment and completion rates',
32      icon: <BarChart2 className="w-8 h-8 text-gray-800" />,
33    }
34  ];
35
36  return (
37    <div className="bg-gray-50 min-h-screen">
38      <div className="bg-white shadow-sm">
39        <div className="max-w-6xl mx-auto px-4 py-4">
40          <h1 className="text-2xl font-bold text-center md:text-left text-gray-900">Admin Dashboard</h1>
41        </div>
42      </div>
43    </div>
```

Рис. В.1.

```

44  { /* Tabs */
45  <div className="max-w-6xl mx-auto px-4 mt-2">
46    <div className="flex space-x-2 justify-center md:justify-start border-b border-gray-200">
47      {tabs.map(tab => (
48        <button
49          key={tab}
50          className={`px-4 py-2 text-sm font-medium rounded-t-lg ${
51            activeTab === tab
52              ? 'bg-white border-t border-l border-r border-gray-200 -mb-px'
53              : 'bg-gray-100 text-gray-600'
54            }`}
55          onClick={() => setActiveTab(tab)}
56        >
57          {tab}
58        </button>
59      )]}
60    </div>
61  </div>
62 </div>
63
64  { /* Content */
65  <div className="max-w-6xl mx-auto p-4 mt-6">
66    <div className="grid grid-cols-1 md:grid-cols-3 gap-4">
67      {actionCards.map(card => (
68        <div
69          key={card.id}
70          className="bg-white p-6 rounded-lg shadow-sm border border-gray-100 flex flex-col items-center text-center hover:shadow-md transition-shadow cursor-pointer"
71        >
72          <div className="mb-4">
73            {card.icon}
74          </div>
75          <h3 className="text-lg font-semibold text-gray-900">{card.title}</h3>
76          {card.subtitle && (
77            <p className="text-sm text-gray-700 mb-0 mt-1">{card.subtitle}</p>
78          )}
79          <p className="text-sm text-gray-600 mt-1">
80            {card.description}
81          </p>
82        </div>
83      )]}
84    </div>
85  </div>
86 </div>
87 );
88 }

```

Рис. В.2.

## ДОДАТОК Г

Код, реалізований в React, який ілюструє модуль проходження курсу

```
1 import { useState } from 'react';
2 import { Play, Pause, Volume2, Maximize, Search, CheckSquare, Square } from 'lucide-react';
3
4 export default function CourseVideoPlayer() {
5   const [isPlaying, setIsPlaying] = useState(false);
6   const [currentTime, setCurrentTime] = useState(0);
7   const [searchQuery, setSearchQuery] = useState("");
8   const [comment, setComment] = useState("");
9   const [courseContent, setCourseContent] = useState([]);
10  const [loading, setLoading] = useState(true);
11  const [error, setError] = useState(null);
12
13  useEffect(() => {
14    const fetchCourseContent = async () => {
15      try {
16        const response = await fetch(`https://api.denaksoy.com/courses/${courseId}/content`);
17        if (!response.ok) {
18          throw new Error("Failed to fetch course content");
19        }
20        const data = await response.json();
21        setCourseContent(data.modules);
22      } catch (err) {
23        setError(err.message);
24      } finally {
25        setLoading(false);
26      }
27    };
28
29    fetchCourseContent();
30  }, [courseId]);
31
32  // Toggle play/pause
33  const togglePlayPause = () => {
34    setIsPlaying(!isPlaying);
35  };
36
37  // Format time (e.g., 0:45)
38  const formatTime = (seconds) => {
39    const minutes = Math.floor(seconds / 60);
40    const remainingSeconds = Math.floor(seconds % 60);
41    return `${minutes}:${remainingSeconds.toString().padStart(2, '0')}`;
42  };
43
44  if (loading) return <p>Loading course content...</p>;
45  if (error) return <p className="text-red-500">Error: {error}</p>;
46
```

Рис. Г.1.

```

47 return (
48   <div className="max-w-5xl mx-auto p-4 bg-gray-50">
49     <div className="grid grid-cols-1 md:grid-cols-3 gap-6">
50       {/* Video Player Column */}
51       <div className="md:col-span-2 space-y-4">
52         {/* Video Player */}
53         <div className="rounded-lg overflow-hidden bg-black relative">
54           {/* Video Placeholder */}
55           <div className="relative aspect-video flex items-center justify-center">
56             <div className="text-white">
57               {isPlaying ? (
58                 <Pause className="w-16 h-16 opacity-0 hover:opacity-100 transition-opacity" />
59               ) : (
60                 <Play className="w-16 h-16" onClick={togglePlayPause} />
61               )}
62             </div>
63           </div>
64
65           {/* Video Controls */}
66           <div className="bg-black text-white p-2 flex items-center space-x-3">
67             <button onClick={togglePlayPause}>
68               {isPlaying ? <Pause className="w-4 h-4" /> : <Play className="w-4 h-4" />}
69             </button>
70
71             {/* Progress Bar */}
72             <div className="flex-1 flex items-center space-x-2">
73               <span className="text-xs">{formatTime(currentTime)}</span>
74               <div className="flex-1 bg-gray-600 h-1 rounded-full">
75                 <div className="bg-white h-1 rounded-full" style={{ width: '20%' }}></div>
76               </div>
77             </div>
78
79             {/* Volume & Fullscreen */}
80             <Volume2 className="w-4 h-4" />
81             <Maximize className="w-4 h-4" />
82           </div>
83         </div>
84

```

Рис. Г.2.

```

85     /* Comments Section */
86     <div className="bg-white p-4 rounded-lg shadow-sm">
87         <h3 className="font-medium text-gray-800 mb-3">Comments</h3>
88         <div className="flex items-start space-x-2">
89             <div className="w-8 h-8 bg-gray-300 rounded-full overflow-hidden flex-shrink-0"></div>
90             <input
91                 type="text"
92                 placeholder="Add a comment..."
93                 className="w-full border border-gray-300 rounded-lg p-2 text-sm"
94                 value={comment}
95                 onChange={(e) => setComment(e.target.value)}
96             >
97         </div>
98     </div>
99 </div>
100
101     /* Course Content & Q&A Column */
102     <div className="space-y-4">
103         /* Course Content */
104         <div className="bg-white p-4 rounded-lg shadow-sm">
105             <h3 className="font-medium text-gray-800 mb-3">Course Content</h3>
106             <div className="space-y-2">
107                 {courseContent.map((item) => (
108                     <div
109                         key={item.id}
110                         className="flex items-center space-x-2 p-2 bg-gray-100 rounded"
111                     >
112                         {item.completed ? (
113                             <CheckSquare className="w-4 h-4 text-gray-700" />
114                         ): (
115                             <Square className="w-4 h-4 text-gray-400" />
116                         )}
117                         <span className="text-sm">{item.title}</span>
118                     </div>
119                 ))
120             </div>
121         </div>
122
123         /* Q&A Section */
124         <div className="bg-white p-4 rounded-lg shadow-sm">
125             <h3 className="font-medium text-gray-800 mb-3">Q&A</h3>
126             <div className="relative">
127                 <input
128                     type="text"
129                     placeholder="Search questions..."

```

Рис. Г.3.