

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

КВАЛІФІКАЦІЙНА РОБОТА
ЗДОБУВАЧА СТУПЕНЯ ВИЩОЇ ОСВІТИ БАКАЛАВР

на тему:

ІНФОКОМУНІКАЦІЙНИЙ ДОДАТОК
ДЛЯ ЗАБЕЗПЕЧЕННЯ ЯКОСТІ НАВЧАННЯ

Міщук Дмитро Сергійович

Київ 2025 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

ЗАТВЕРДЖУЮ

Завідувачка кафедри ІТ

д.т.н., доцент Гончаренко Т.А.

„_____” _____ 2025 року

**КВАЛІФІКАЦІЙНА РОБОТА
ЗДОБУВАЧА СТУПЕНЯ ВИЩОЇ ОСВІТИ БАКАЛАВР**

на тему: **«ІНФОКОМУНІКАЦІЙНИЙ ДОДАТОК
ДЛЯ ЗАБЕЗПЕЧЕННЯ ЯКОСТІ НАВЧАННЯ»**

Я як здобувач вищої освіти КНУБА розумію і підтримую політику закладу з академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) незгоду чи допомогу під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Здобувач

Міщук Дмитро Сергійович

122 «Комп'ютерні науки»

(спеціальність)

Інформаційні управляючі системи і технології

(освітня програма)

Групи КН-21

Керівник Рябчун Ю.В.

(прізвище та ініціали)

Доктор філософії

(вчене звання, науковий ступінь)

Рецензент к.т.н., доц. Доля О.В.

(Прізвище та ініціали)

Ідентичність підтверджую

Київ, 2025 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БУДІВНИЦТВА І
АРХІТЕКТУРИ**

Факультет:	Автоматизації інформаційних технологій
Випускова кафедра:	Інформаційних технологій
Освітній ступінь:	Бакалавр
Спеціальність:	Комп'ютерні науки
Освітня програма:	Інформаційні управляючі системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТ

Тетяна ГОНЧАРЕНКО

„___” _____ 2025 року

ЗАВДАННЯ

ДО ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ НА
ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВР

Здобувач	Міщук Дмитро Сергійович
1. Тема роботи ІНФОКОМУНІКАЦІЙНИЙ ДОДАТОК ДЛЯ ЗАБЕЗПЕЧЕННЯ ЯКОСТІ НАВЧАННЯ	
затверджена наказом ректора КНУБА №235/23/25 від «14» лютого 2025 року	
2. Керівник роботи	Рябчун Юлія Володимирівна, PhD

3. Строк подання Здобувачем роботи до захисту 24.05.2025

4. Зміст пояснювальної записки за розділами:

P.1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

P.2 ПРОЄКТУВАННЯ ІНФОКОМУНІКАЦІЙНОГО ДОДАТКА

P.3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ

P.4 АНАЛІЗ РЕЗУЛЬТАТІВ ТА ПЕРСПЕКТИВИ РОЗВИТКУ

5. Графічний матеріал за розділами:

P.1 _____

P.2 _____

P.3 _____

P.4 _____

6. Календарний план виконання роботи:

Види робіт та їх зміст	Дата виконання
Розділ 1	17.01.2025
Розділ 2	27.02.2025
Розділ 3	21.04.2025
Розділ 4	20.05.2025
Остаточне оформлення роботи	24.05.2025
Направлення роботи для перевірки на плагіат	24.05.2025
Попередній захист роботи на випусковій кафедрі	26.05.2025
Направлення роботи на рецензування	26.05.2025

7. Консультанти розділів атестаційної випускної роботи

Розділ	Прізвище, ініціали та посада консультанта	Перевірив	
		дата	підпис
Розділ 1	Рябчун Ю.В., доц.каф.ІТ	23.01.2025	
Розділ 2	Рябчун Ю.В., доц.каф.ІТ	27.02.2025	
Розділ 3	Рябчун Ю.В., доц.каф.ІТ	21.05.2025	
Розділ 4	Рябчун Ю.В., доц.каф.ІТ	21.05.2025	

8. Дата видачі завдання листопад 2024 р.

Зав. кафедри			Гончаренко Т.А.
	(підпис)		(прізвище та ініціали)
Керівники			Рябчун Ю.В.
	(підпис)		(прізвище та ініціали)
Здобувач			Міщук Д.М.
	(підпис)		(прізвище та ініціали)

РЕЗЮМЕ (SUMMARY) <i>до кваліфікаційної випускної роботи Здобувача:</i>	Міщук Дмитро Сергійович Mishchuk Dmitry		
<i>ЗВО</i>	Київський національний університет будівництва і архітектури		
<i>Тема (українською та англійською)</i>	Інфокомунікаційний додаток для забезпечення якості навчання Infocommunication Application for Ensuring the Quality of Education		
<i>Освітній ступінь</i>	Бакалавр		
<i>Факультет</i>	Автоматизації і інформаційних технологій		
<i>Випускова кафедра</i>	Інформаційних технологій		
<i>Спеціальність</i>	122 «Комп'ютерні науки»		
<i>Освітня програма</i>	Інформаційні управляючі системи та технології		
<i>Керівник</i>	Рябчун Юлія Володимирівна		
<i>Обсяг роботи:</i>	пояснювальна записка, стор.	розділів	Кількість слайдів презентації
	86	4	
<i>Розділ 1.</i>	АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ		
<i>Розділ 2.</i>	ПРОЄКТУВАННЯ ІНФОКОМУНІКАЦІЙНОГО ДОДАТКА		
<i>Розділ 3.</i>	РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ		
<i>Розділ 4.</i>	АНАЛІЗ РЕЗУЛЬТАТІВ ТА ПЕРСПЕКТИВИ РОЗВИТКУ		
<i>Ключові слова:</i> <i>Keywords:</i>	інфокомунікаційні технології, навчальний додаток, дистанційне навчання, клієнт-серверна архітектура, база даних, гейміфікація, JavaScript. infocommunication technologies, educational application, distance learning, client-server architecture, database, gamification, JavaScript		

АНОТАЦІЯ

Міщук Д.С. Інфокомунікаційний додаток для забезпечення якості навчання.

Атестаційна випускна робота бакалавра за спеціальністю 122 «Комп'ютерні науки», освітня програма «Інформаційні управляючі системи та технології». – Київський національний університет будівництва та архітектури. – Київ, 2025.

Дипломна робота присвячена розробці інфокомунікаційного додатку, що має на меті підвищення якості освітнього процесу шляхом інтеграції сучасних цифрових технологій у навчальне середовище. У роботі виконано аналіз існуючих освітніх платформ, виявлено їхні основні недоліки та сформульовано вимоги до нового інструменту, який поєднує функціональність управління курсами, інтерактивне навчання, відеозв'язок, тестування, гейміфікацію та аналітику успішності.

Розроблена система базується на клієнт-серверній архітектурі з використанням технологій Node.js, MySQL, HTML/CSS, JavaScript та REST API. Особливу увагу приділено безпеці даних, зручності інтерфейсу та можливостям персоналізації навчального процесу. Реалізовано механізми реєстрації користувачів, управління навчальними матеріалами, проведення онлайн-уроків, автоматичного оцінювання, інтерактивного чату та навчальних ігор. Отриманий додаток має значний потенціал для подальшого вдосконалення та масштабування в умовах сучасної цифрової трансформації освіти в Україні.

Робота викладена на 86 аркушах, 23 рисунків, використаної літератури із 32 найменувань.

Ключові слова: інфокомунікаційні технології, навчальний додаток, дистанційне навчання, клієнт-серверна архітектура, база даних, гейміфікація, JavaScript.

SUMMARY

Mishchuk D.S. Infocommunication Application for Ensuring the Quality of Education.

Bachelor's Degree Qualification Work in the specialty 122 "Computer Science", educational program "Information Control Systems and Technologies". – Kyiv National University of Construction and Architecture. – Kyiv, 2025.

The thesis is dedicated to the development of an infocommunication application aimed at improving the quality of the educational process by integrating modern digital technologies into the learning environment. The paper includes an analysis of existing educational platforms, identifies their main shortcomings, and formulates the requirements for a new tool that combines course management functionality, interactive learning, video communication, testing, gamification, and performance analytics.

The developed system is based on a client-server architecture using technologies such as Node.js, MySQL, HTML/CSS, JavaScript, and REST API. Special attention is given to data security, user interface convenience, and the ability to personalize the learning process. The system implements user registration, management of learning materials, conducting online lessons, automatic grading, interactive chat, and educational games. The resulting application has significant potential for further improvement and scaling within the context of the ongoing digital transformation of education in Ukraine.

The paper consists of 85 pages, includes 23 figures, and references 32 sources.

Keywords: infocommunication technologies, educational application, distance learning, client-server architecture, database, gamification, JavaScript.

ЗМІСТ

ВСТУП	9
Розділ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	11
1.1 Аналіз та актуальність використання інфокомунікаційних додатків у навчанні	12
1.2 Основні проблеми вже існуючих навчальних додатків	13
1.3 Інноваційні рішення до проєктування навчальних додатків	19
1.4 Формування вимог і концептуальна модель проєкту	22
1.5 Постановка задачі та очікувані результати	26
Розділ 2. ПРОЄКТУВАННЯ ІНФОКОМУНІКАЦІЙНОГО ДОДАТКА	28
2.1. Архітектура програмної системи	28
2.1.1. Огляд можливих архітектурних підходів	28
2.1.2. Обґрунтування вибору архітектури	29
2.1.3. Компонентна структура системи	30
2.1.4. Взаємодія між компонентами	31
2.2. Проєктування бази даних	31
2.2.1. Визначення основних сутностей та атрибутів	31
2.2.2. Структура бази даних та її зв'язки	36
2.2.3. Оптимізація та нормалізація даних	37
2.2.4. Вибір системи керування базами даних (СКБД)	38
2.3. Вибір технологій для реалізації додатку	39
2.3.1. Обґрунтування вибору мови програмування	39
2.3.2. Використання фреймворків і бібліотек	41
2.3.3. Архітектура серверної інфраструктури	42
2.3.4. Засоби розробки та тестування	43
2.4. Інтерфейс та користувацький досвід (UI/UX)	44

	9
2.4.1. Значення UI/UX у цифровій освіті	44
2.4.2. Цілі та завдання UI/UX у навчальних середовищах	45
2.4.3. Основні принципи UX/UI-дизайну для освітнього додатку	46
Розділ 3. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ	49
3.1. Загальний опис реалізації	49
3.2. Реалізація клієнтської частини (інтерфейсу)	54
3.3. Інструкція користувача системи	63
3.4. Тестування та перевірка функціональності	65
3.5. Переваги розробленого додатку	68
Розділ 4. АНАЛІЗ РЕЗУЛЬТАТІВ ТА ПЕРСПЕКТИВИ РОЗВИТКУ	72
4.1 Оцінка ефективності реалізованого рішення	72
4.2. Можливості масштабування та інтеграції з іншими системами	75
4.3 Пропозиції щодо майбутніх удосконалень	76
ВИСНОВОК	80
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	82

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Node.js — середовище виконання JavaScript, яке дозволяє запускати код на сервері.

MySQL — система керування реляційними базами даних, яка використовується для збереження та обробки інформації.

HTML/CSS — HTML (мова розмітки гіпертексту) створює структуру веб-сторінки, а CSS (каскадні таблиці стилів) відповідає за її зовнішній вигляд.

JavaScript — це мова програмування, що забезпечує динамічність і інтерактивність веб-сторінок.

REST API — це архітектурний стиль, який дозволяє клієнту та серверу обмінюватися даними через HTTP-запити.

ВСТУП

Сучасний світ переживає етап швидкої інформатизації, і ця зміна активно відображається на освітньому процесі. Онлайн-навчання вже стало важливим компонентом сучасної освіти, і його значення продовжує зростати як в Україні, так і у світі загалом. Сьогодні існує безліч платформ та застосунків для навчання, які забезпечують доступ до знань, покращують процес навчання та сприяють розвитку учнів. Однак з часом багато з цих платформ втрачають свою ефективність і потребують оновлення, оскільки технології швидко змінюються, а вимоги користувачів еволюціонують.

Незважаючи на широкий вибір існуючих рішень, багато з них не відповідають сучасним вимогам, а також не враховують новітніх технологій, які можуть значно покращити якість навчання. Розробка інфокомунікаційного додатку для забезпечення якості навчання є важливим кроком у створенні інноваційного та ефективного інструменту, який допоможе оптимізувати освітній процес. Такий додаток, який враховує потреби сучасного навчання та інтегрує найновіші технології, стане не лише зручним і ефективним для користувачів, а й створить умови для адаптації до швидко змінюваного освітнього середовища. Платформа, що поєднує всі ці характеристики, стане більш інклюзивною, сучасною та доступною, що дозволить підвищити ефективність навчання та взаємодії між усіма учасниками освітнього процесу.

Результатом розробки стане ефективна платформа, яка забезпечить підвищення якості освіти за рахунок інтерактивних технологій, задовольнить потреби різних категорій користувачів, забезпечить зручність і гнучкість у навчанні та знизить бар'єри у доступі до освіти. Таким чином, запропоноване рішення сприятиме вдосконаленню освітнього процесу, зробивши його доступним і ефективним для широкого кола користувачів.

Метою даної кваліфікаційної роботи є розробка інфокомунікаційного додатку, що стане потужним інструментом для забезпечення якості навчання, враховуючи сучасні тенденції та інноваційні підходи. Цей додаток сприятиме

підвищенню ефективності навчального процесу через інтеграцію сучасних технологій та забезпечення комфортного користування для всіх учасників.

Завдання дослідження:

1. Аналіз сучасних існуючих інфокомунікаційних рішень для освітнього процесу.
2. Формування вимог до інфокомунікаційного додатка.
3. Розробка концепції інфокомунікаційного додатку.
4. Розробка інфокомунікаційного додатку.
5. Впровадження та тестування застосунку.

Об'єкт дослідження – процес забезпечення якості навчання в освітньому середовищі з використанням сучасних інформаційно-комунікаційних технологій.

Предмет дослідження – методи та засоби розробки інфокомунікаційного додатка, спрямованого на підвищення якості навчального процесу, включаючи функціональні можливості для оптимізації комунікації, управління навчальним контентом, моніторингу успішності та покращення взаємодії між учасниками освітнього процесу.

Розділ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

Онлайн-навчання, на сьогоднішній день, стало невід'ємною частиною освітнього процесу в усьому світі. Потреба освітніх платформ значно зросла з початком пандемії COVID-19, що підштовхнуло розвиток дистанційної освіти. З часом це питання набуло особливої важливості, зважаючи на виклики, пов'язані з повномасштабною війною Російської Федерації проти України, розвитком гібридних форматів навчання та прагненням зробити освітні можливості максимально зручними та доступними для сучасного суспільства. Проте, незважаючи на існування великої кількості навчальних застосунків, їх функціональність часто залишається обмеженою та малоефективною, а інтерфейси – складними для користувачів. Тому, існує потреба у розробці та вдосконаленні інфокомунікаційних додатків, які сприятимуть підвищенню якості освіти.

Аналіз предметної області є важливим етапом розробки будь-якого програмного забезпечення, оскільки дозволяє краще зрозуміти поточний стан проблеми, основні виклики та потреби користувачів. У даному випадку аналіз допомагає виявити недоліки існуючих навчальних платформ і визначити області, які потребують подальшого удосконалення, а також визначити потреби користувачів, а саме: учнів, викладачів та адміністративного персоналу. Аналіз також створює основу для можливого впровадження інновацій, приміром використання новітніх технологій, наприклад, штучний інтелект та адаптивні системи. Також, цей етап дозволяє визначити напрямки підвищення ефективності навчання завдяк інтерактивності, персоналізації, зручності користування та інклюзивності. Таким чином, аналіз предметної області є базою для обґрунтування вибору функціоналу, структури та технологій, які будуть застосовані у розробці застосунку.

У розділі представлено аналіз актуальності використання інфокомунікаційних додатків у навчанні, розглянуто проблеми існуючих платформ

та запропоновано інноваційні рішення для їх удосконалення. Визначено основні вимоги до навчальних додатків та сформульовано концептуальну модель проєкту. Сформовано постановку задачі з ключовими цілями, що мають бути досягнуті за допомогою розробленого програмного забезпечення, зокрема підвищення ефективності та зручності навчання.

1.1 Аналіз та актуальність використання інфокомунікаційних додатків у навчанні

Сучасний світ неможливо уявити без інформаційних технологій, які стали невід'ємною частиною багатьох сфер діяльності, включно з освітою. Особливої актуальності вони набули у періоди глобальних викликів, таких як пандемія COVID-19, яка змусила освітні заклади по всьому світу перейти на дистанційну форму навчання [1, 2]. Інфокомунікаційні додатки, такі як Zoom [3], Google Classroom [4], Microsoft Teams [5, 6], стали головним інструментом для підтримання навчального процесу. Вони забезпечили можливість комунікації між викладачами та студентами, доступ до навчальних матеріалів і проведення занять у форматі реального часу, коли традиційне навчання стало неможливим. Попри зменшення піку пандемії і повернення до звичайного життя, актуальність дистанційного навчання не втрапилася. У 2022 році внаслідок повномасштабного вторгнення Росії проти України дистанційне навчання знову стало незамінним інструментом. Воно дозволило внутрішньо переміщеним особам, біженцям та людям на тимчасово окупованих територіях зберегти доступ до освітніх послуг незалежно від місцезнаходження.

Окрім кризових обставин, дистанційне навчання за допомогою інфокомунікаційних додатків стало одним із найбільш зручних методів здобуття освіти [7]. Воно дозволяє економити час на дорогу, забезпечує рівний доступ до якісної освіти незалежно від географічного розташування, створює можливості для навчання в пріоритетних навчальних закладів або курсів незалежно від їхнього місцезнаходження. Самі ж інфокомунікаційні додатки дозволяють

створювати інтерактивні уроки, тестування, а також підтримують групові проєкти та обговорення. Це сприяє розвитку навичок співпраці, критичного мислення та аналізу, і значно спрощують комунікацію між учнями та викладачами. Однією з важливих переваг таких застосунків є мінімізація потреби в друкованих матеріалах, це не лише зумовлює економію коштів, а й допомагає дбати про довкілля, а у сучасному світі прагнення до екологічності стає одним із ключових пріоритетів, і такі рішення гармонійно вписуються в цю глобальну тенденцію. І звісно цифрові платформи зберігають час, оскільки забезпечують миттєвий доступ до інформації та автоматичну перевірку завдань.

Саме тому актуальність використання інфокомунікаційних додатків у навчанні продовжує дедалі зростати. Вони стали не лише інструментами для подолання кризових ситуацій, але й платформами для інноваційного підходу до здобуття освіти як у традиційній, так і в дистанційній формі навчання. Водночас розвиток цих додатків, адаптація до нових технологій і постійне вдосконалення є необхідними для забезпечення високої якості освіти та відповідності сучасним потребам.

1.2 Основні проблеми вже існуючих навчальних додатків

Сучасний освітній процес дедалі більше покладається на використання інфокомунікаційних додатків. Серед найпопулярніших платформ, які широко застосовуються в навчанні, можна виділити Zoom, Skype, Google Classroom, Moodle, Microsoft Teams, тощо. Вони забезпечують базові можливості для дистанційного навчання: відеоконференції, доступ до навчальних матеріалів, виконання завдань і тестування. Однак попри їхню популярність, існує низка проблем, які ускладнюють їх ефективне використання в освітньому процесі.

Багато платформ використовуються не за призначенням. Наприклад, Zoom (рис.1.1), що спочатку був розроблений як інструмент для бізнес-зустрічей, не забезпечує достатньої інтерактивності для освітніх цілей. Ця платформа має більше недоліків, ніж переваг, якщо розглядати її з точки зору освітніх потреб.

Одним із ключових недоліків Zoom є складність у використанні [3, 8]. Процес доступу до платформи через ідентифікатори та паролі часто викликає труднощі, особливо у дітей, для яких це може стати значною перешкодою; відсутність спеціальних освітніх функцій, оскільки платформа не надає доступу до інтегрованих навчальних матеріалів, інструментів для тестування чи інших функцій, необхідних для організації сучасного освітнього процесу. Крім того, безкоштовна версія Zoom обмежує час однієї конференції до 40 хвилин, що створює незручності як користувачів. Це змушує користувачів або переривати уроки для повторного входу, або переходити на платну версію, що не завжди є можливим чи доцільним. Попри те, що Zoom набув значного поширення в Україні, його використання в освітніх закладах часто не відповідає початковому призначенню платформи.

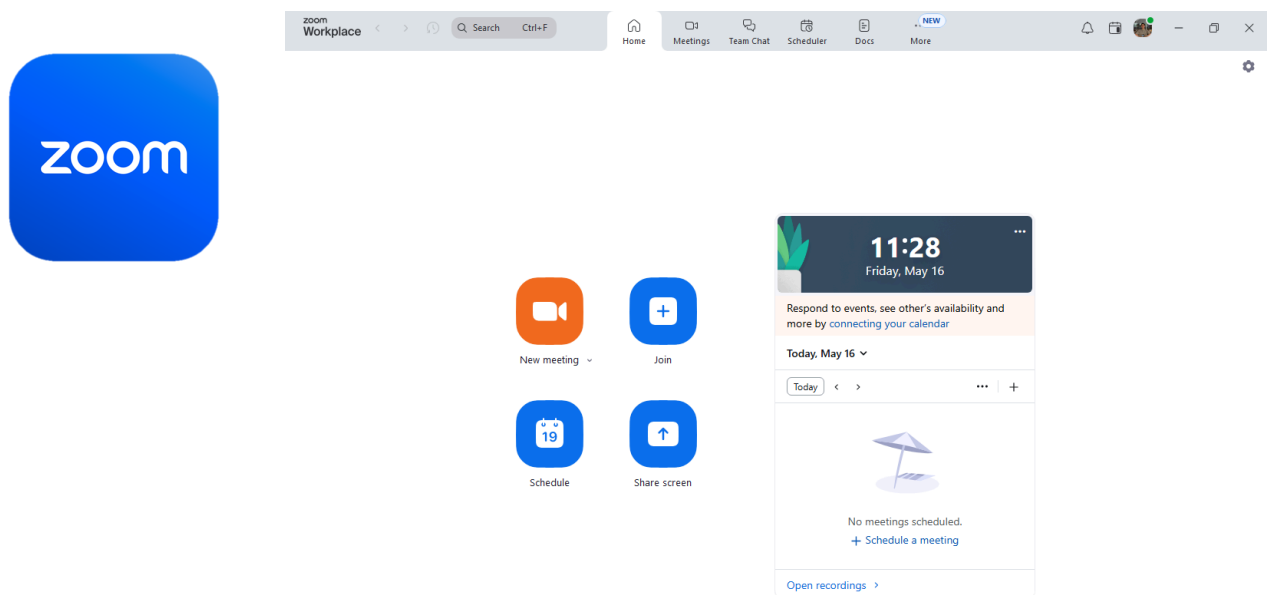


Рисунок 1.1. Фрагмент програми Zoom для організації відеоконференцій

Також до вищезазначених проблем можна віднести використання застосунку Skype (рис.1.2), який за своїм функціоналом має багато спільного із Zoom і стикається з подібними недоліками [9]. Skype також не забезпечує інтеграції

освітніх матеріалів, інструментів для тестування або інших функцій, необхідних для організації навчального процесу. Якість зв'язку в Skype часто залежить від стабільності інтернет-з'єднання, і при низькій швидкості зв'язку виникають проблеми зі звуком або відео.

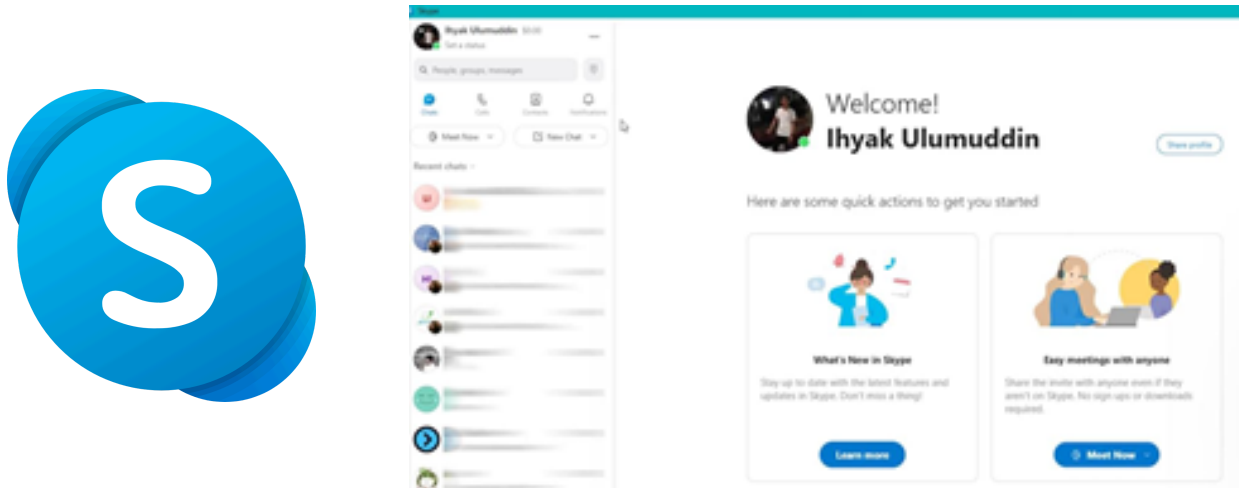


Рисунок 1.2. Skype сервіс для організації відеозустрічей та спілкування

Хоча Skype пропонує деякі базові функції для спілкування, такі як чат і відеодзвінки, вони є недостатніми для повноцінного забезпечення освітніх потреб. Тому через обмежений функціонал та численні недоліки Zoom, як і Skype не можуть повністю задовольнити потреби сучасного навчального процесу.

Google Classroom (рис.1.3) має більше переваг у порівнянні з попередніми додатками, адже він був розроблений спеціально для освітніх цілей і пропонує досить зручний інтерфейс для організації навчального процесу. Однак цей застосунок також має свої недоліки, які обмежують його функціональність [4, 10].

Одним із головних мінусів Google Classroom є відсутність вбудованої функції відеоконференцій. Це змушує здобувачів освіти та викладачів використовувати додаткові програми, для проведення онлайн-занять, що ускладнює процес навчання та створює незручності для користувачів. І хоча Classroom дозволяє обмінюватися завданнями, матеріалами та зворотним зв'язком, він не пропонує інтерактивних інструментів, таких як віртуальні дошки, живі опитування чи інтегровані тести в реальному часі, що могли б зробити навчання

більш динамічним і зручнішим. Також варто зазначити складність у налаштуванні для нових користувачів. Учням і навіть деяким викладачам може бути важко зрозуміти функції системи без попереднього навчання або інструкцій, що ускладнює її швидке впровадження.

Тому ці недоліки свідчать про необхідність його вдосконалення, особливо з урахуванням сучасних вимог до дистанційного навчання.

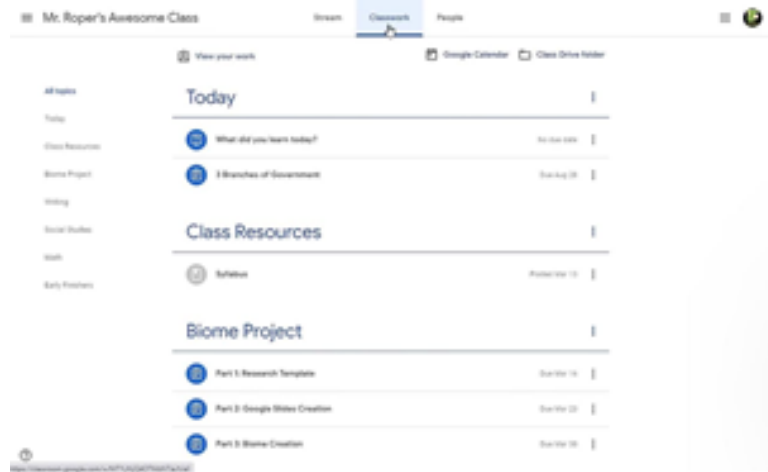


Рисунок 1.3. Google Classroom вебсервіс від Google для спрощення управління навчальними завданнями в електронному форматі

Moodle (рис.1.4) є також досить поширеною серед освітніх платформ в Україні, яка пропонує широкий спектр можливостей для організації навчального процесу, таких як завантаження додаткових матеріалів, проведення тестувань та оцінювання роботи учнів [11, 12].



Рисунок 1.4. Moodle - відкрита система управління навчанням

Проте вона також має низку суттєвих недоліків, які знижують її ефективність та зручність використання. Також немає вбудованої функції для відеоконференцій, що змушує освітні заклади використовувати додаткові інструменти для проведення онлайн-занять. Moodle має складний і застарілий інтерфейс, який часто перевантажений і робить навігацію платформою незручною та інтуїтивно незрозумілою. Для повного освоєння всіх функцій Moodle потрібен тривалий час і спеціальне навчання, що не завжди є можливим у швидкому темпі освітнього процесу.

Microsoft Teams (Рис.1.5) [5, 6, 13, 14] є однією з найбільш функціональних платформ для освітніх цілей, на сьогоднішній день, яка забезпечує широкий спектр можливостей, таких як інтеграція з іншими сервісами Microsoft, доступ до навчальних матеріалів, безпеку даних та тестування. Цифрова платформа дає можливість здобувачам освіти звернутися до викладача з будь-якими питаннями та отримати швидку відповідь. Проте, попри ці переваги, платформа потребує подальшого вдосконалення та адаптації до потреб сучасного користувача.

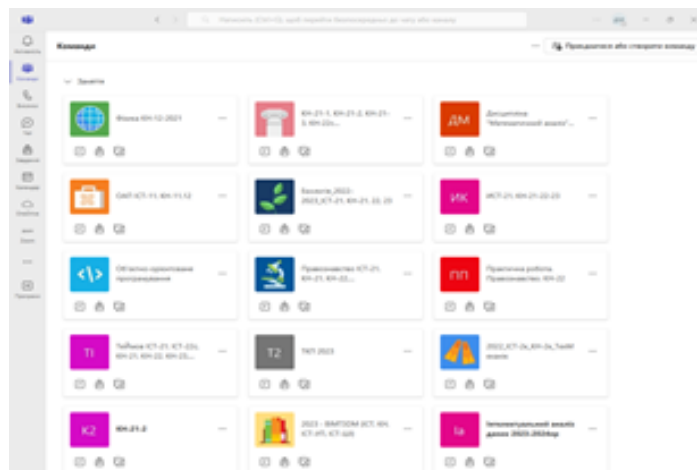


Рисунок 1.5. Microsoft Teams корпоративна платформа

Однією з головних проблем Microsoft Teams є висока вимогливість до технічних ресурсів пристрою [5, 6, 13 - 15]. Платформа часто споживає значну

кількість оперативної пам'яті, що спричиняє збої чи уповільнює роботу на старих або малопотужних пристроях, які досі активно використовуються багатьма навчальними закладами та користувачами. Іншим суттєвим недоліком є складність інтеграції для новачків. Викладачі та студенти, які раніше не працювали з продуктами Microsoft, можуть зіткнутися з труднощами у налаштуванні облікових записів, пошуку функцій чи організації процесу навчання. Платформа також має обмеження у доступі для користувачів, які не мають корпоративної ліцензії. Навчальним закладам необхідно придбати пакети Microsoft 365, що може стати фінансовим тягарем, особливо для шкіл та університетів із обмеженим бюджетом. Додатково, інтерфейс Microsoft Teams може бути надмірно складним для використання в освітньому середовищі, особливо для молодших учнів. Велика кількість вкладок, меню та функцій іноді створює враження перевантаженості, що ускладнює швидку навігацію платформою.

Загалом, хоча Microsoft Teams є потужним інструментом, його технічна вимогливість, фінансові витрати та складність для нових користувачів свідчать про необхідність її адаптації до потреб сучасної освіти.

Окрім технічних проблем, які були зазначені, в Україні відсутня єдина платформа для навчання, затверджена та рекомендована Міністерством освіти на державному рівні. Це створює значні труднощі для освітнього процесу. Наприклад, у державному секторі успішно функціонує додаток «Дія» [16], який є прикладом універсального та зручного інструменту, що відповідає потребам сучасних користувачів. Однак в освітній сфері досі немає аналогічного рішення, яке б забезпечувало уніфіковану систему для всіх навчальних закладів. У зв'язку з цим Україні доцільно розробити власну національну платформу для навчання, яка була б офіційно затверджена на законодавчому рівні. Така платформа повинна враховувати сучасні технологічні можливості й бути універсальною і могла б стати єдиним стандартом, який спростив би перехід учнів та студентів між навчальними закладами, а також створив би єдине середовище для комунікації,

тестування, зберігання навчальних матеріалів та проведення уроків у будь-якому форматі.

Підсумовуючи, варто зазначити, що застосунки, які сьогодні є досить поширеними в Україні для освітніх цілей, або є недоречними у використанні, або потребують значного оновлення. Їм бракує впровадження сучасних технологій, інноваційних підходів і врахування потреб сучасного користувача. Усі виявлені проблеми цих платформ необхідно ретельно проаналізувати та врахувати при подальшій розробці нового інфокомунікаційного додатка, який відповідатиме сучасним вимогам і сприятиме підвищенню якості освітнього процесу.

1.3 Інноваційні рішення до проєктування навчальних додатків

Будь-які інноваційні рішення спрямовані на досягнення позитивних результатів, які покращують та спрощують життя користувачів. Інновації у сфері навчальних додатків повинні мати чітко визначену мету та відповідати сучасним технологічним можливостям. Вони дозволяють змінити підходи до освіти, роблячи навчальний процес більш інтерактивним, доступним та ефективним.

Зважаючи на це, доцільним є створення в Україні єдиної національної освітньої платформи, офіційно затвердженої на законодавчому рівні. Така платформа повинна:

- враховувати сучасні технології;
- бути універсальною для користувачів різного віку;
- забезпечувати зручність у використанні для вчителів та учнів;
- інтегруватися з наявними освітніми системами.

Крім того, створення такої платформи сприятиме стандартизації освітнього процесу, зокрема спрощенню переходу між навчальними закладами та наданню єдиного середовища для комунікації, тестування, зберігання матеріалів та проведення занять. Для створення ефективного навчального застосунку з інноваційними підходами необхідно врахувати досвід роботи з уже існуючими платформами, адже аналізуючи їх переваги та недоліки, можна розробити новий

продукт, що враховуватиме сучасні технологічні можливості та потреби користувачів. Наш застосунок, не лише усуватиме недоліки попередніх рішень, але й впроваджуватиме *інноваційні елементи*, що зроблять навчальний процес більш інтерактивним, доступним і зручним.

Хмарні технології. Розглянувши проблему перевантаження пристроїв через великий обсяг навчальних матеріалів, хмарні технології пропонують ефективне рішення цієї проблеми, забезпечуючи обчислювальні ресурси для обробки та зберігання даних за принципом сервісу [17, 18]. Це означає, що користувачам не потрібно інвестувати в дороге комп'ютерне обладнання — технологія адаптована для роботи на звичайних гаджетах чи ПК через спеціальний додаток. Інтеграція хмарних технологій у навчальні програми зробила б їх доступнішими для широкого кола користувачів, незалежно від технічних можливостей їхніх пристроїв. Це особливо актуально для студентів і освітніх закладів із обмеженими ресурсами.

Водночас, ключовим недоліком цієї технології залишається необхідність постійного доступу до інтернету. Для вирішення цього питання доцільно реалізувати функцію завантаження матеріалів. Таким чином, користувачі можуть самостійно обирати та зберігати лише потрібні файли, що забезпечить доступ до них навіть без підключення до мережі. Хмарні технології вже довели свою ефективність і активно використовуються у різних сферах. Їхнє впровадження у навчальні додатки значно підвищило б зручність і функціональність освітніх процесів.

Штучний інтелект (AI). Наразі штучний інтелект стрімко завоював популярність серед користувачів, завдяки своїй здатності адаптуватися до різних сфер життя. Вже існують численні приклади успішної інтеграції AI в сучасні застосунки, як-от Photoshop AI [19, 20], який завдяки розумним алгоритмам значно спрощує та пришвидшує роботу з графічними зображеннями. У нашому контексті AI відкриває ще ширші горизонти можливостей. Серед найперспективніших ідей:

- *Персоналізоване навчання:* AI аналізує прогрес студента та пропонує індивідуальні матеріали, завдання чи тести, враховуючи його сильні та слабкі сторони.
- *Розумні асистенти:* AI-боти допомагають студентам, відповідаючи на запитання, пояснюючи складні теми та надаючи додаткові ресурси для поглиблення знань.
- *Інтерактивний голосовий ввід:* голосові функції для швидкого пошуку інформації, створення нотаток чи постановки запитань.
- *Генерація тестів і текстів:* AI може автоматично створювати завдання різного рівня складності, а також стискати великі тексти у ключові тези, економлячи час студентів.
- *Мовна підтримка:* AI здатний перекладати навчальні матеріали та пропонувати інтерактивні завдання для практики іноземних мов.
- *Аналіз навчальних даних:* опрацювання великих обсягів інформації дозволяє викладачам бачити загальну картину успішності та надавати точний зворотний зв'язок.

Гейміфікація. Сучасна система освіти переживає кризу відсутності мотивації довивчення різних дисциплін. Вихід з такої ситуації, до певної міри, полягає у використанні інноваційних технологій у системі навчання й виховання, у формуванні мотивації, інтересу, у наданні інформації цікавими нетрадиційними способами. Одним з таких способів є гейміфікація [21, 22]. Ігри – унікальне явище. Вони не просто мотивують грати, але й тривалий час підтримують цю мотивацію. За статистикою 97% дітей, віком від 12 до 17 років грають у відеоігри [23, 24]. Така наполегливість – мрія викладачів, адже захопити учнів на тривалий час – це справжнє мистецтво. Гейміфікація – це «впровадження ігрових технологій в неігрові процеси, в тому числі в освіту», а також «використання ігрової механіки, естетики та ігрового мислення для залучення людей в навчання і рішення різних задач та для підвищення їх мотивації» [25]. Це дозволяє непомітно освоювати те, що було складним для засвоєння раніше. Ігрові елементи сприяють

підвищенню мотивації учнів і роблять навчання захопливим. Завдяки механікам змагань, нагород і інтерактивних завдань учні залучаються до освітнього процесу триваліший час. Гейміфікація дозволяє інтегрувати складні теми у просту ігрову форму, роблячи їх легшими для засвоєння. Тому така технологія стає потужним інструментом для подолання кризи зацікавленості, адже вона поєднує розвагу та навчання, сприяючи ефективному засвоєнню знань.

Сучасна система освіти потребує інноваційних рішень, які сприятимуть підвищенню якості навчального процесу. Успішна розробка нових освітніх додатків має базуватися на врахуванні наявних проблем, впровадженні новітніх технологій і адаптації їх до реальних потреб користувачів. Такий підхід забезпечить створення ефективного, доступного й універсального продукту для розвитку української освіти.

1.4 Формування вимог і концептуальна модель проєкту

Для створення ефективного додатка необхідно ретельно сформулювати вимоги до його функціональності, дизайну, безпеки та технологій.

1. Реєстрація та управління користувачами

- 1.1. Реєстрація користувачів з різними ролями: учні, вчителі, адміністратори.
- 1.2. Можливість зміни ролі користувача адміністратором.
- 1.3. Відновлення доступу до аккаунта (через email або SMS).
- 1.4. Можливість групової реєстрації (наприклад, імпорт учнів у клас).

2. Управління розкладом та заняттями

- 2.1. Створення та редагування розкладу занять: інтерактивний календар із зазначенням часу, предмету та вчителя.
- 2.2. Сповіщення про початок уроку.
- 2.3. Архів минулих занять із записами відеоуроків та матеріалів.

3. Проведення уроків

- 3.1. Вбудований інструмент для відеоконференцій:
 - Підтримка відео, аудіо та текстових чатів.

- Демонстрація екрану.
- Віртуальна дошка для спільного використання.

3.2. Підтримка роботи з інтерактивними тестами та опитуваннями під час уроку, вікторинами у форматі гри, щотижневі челенджі для підтримки інтересу учнів.

3.3. Можливість запису уроку для подальшого перегляду.

3.4. Функція «підняти руку» для запитань від учнів.

4. Управління навчальними матеріалами

4.1. Завантаження та зберігання навчальних матеріалів:

- Підтримка файлів формату PDF, DOCX, PPT, MP4 тощо.
- Організація матеріалів за предметами або темами.

4.2. Доступ до матеріалів за правами (тільки для вчителів/учнів).

4.3. Пошук за назвою чи ключовими словами.

4.4. Хмарне зберігання з обмеженням обсягу для кожного користувача.

5. Тестування та контроль знань

5.1. Автоматизоване створення тестів:

• Тести з вибором відповідей, відкритими питаннями, встановленням відповідності.

5.2. Автоматична перевірка тестів і виставлення оцінок.

5.3. Статистика результатів тестів для учнів і вчителів.

5.4. Збереження історії тестування для аналізу успішності.

6. Електронний журнал та оцінювання

6.1. Електронний журнал для вчителів: внесення оцінок, відвідуваності, коментарів.

6.2. Доступ учнів і батьків до оцінок через особистий кабінет.

7. Чати та обмін повідомленнями

7.1. Чати для класів, груп і індивідуальних користувачів.

7.2. Надсилання мультимедійних файлів (зображення, аудіо, відео).

7.3. Можливість модерації чатів адміністраторами.

8. Захист даних та конфіденційність

8.1. Шифрування персональних даних користувачів.

8.2. Аутентифікація через двоетапну перевірку.

9. Налаштування системи

9.1. Можливість зміни мови інтерфейсу (українська, англійська тощо) та перекладу матеріалів іншими мовами.

9.2. Налаштування сповіщень для кожного користувача.

9.3. Можливість голосового введення/керування.

Нефункціональні вимоги

1. Продуктивність

1.1 Час відгуку на будь-яку дію користувача не повинен перевищувати 2 секунд.

1.2 Система повинна обробляти до 10 000 одночасних користувачів без зниження продуктивності.

2. Надійність

2.1 Доступність системи повинна становити не менше 99,9% часу протягом року.

2.2 Автоматичне резервне копіювання даних повинно виконуватись щодня.

3. Безпека

3.1 Захист даних користувачів за стандартами GDPR.

3.2 Зашифрована передача даних за допомогою протоколів SSL/TLS.

3.3 Підтримка двофакторної аутентифікації.

4. Юзабіліті

4.1 Інтуїтивно зрозумілий адаптивний дизайн для різних пристроїв.

4.2 Підтримка різних мов у системі.

5. Інтеграція

5.1 Наявність API для інтеграції з іншими платформами

5.2 Підтримка імпорту та експорту даних у стандартних форматах (CSV, JSON).

6. Законодавчі вимоги

6.1 Відповідність українським стандартам захисту персональних даних.

6.2 Освітні матеріали повинні відповідати нормам авторського права.

Концептуальна модель (рис.1.6) проєкту, що розробляється, має охоплювати ключові аспекти розробки, функціонування та впровадження додатка, спрямованого на підвищення ефективності навчального процесу. У моделі враховуються основні компоненти, взаємозв'язки між ними, цільова аудиторія, технічні засоби, а також інноваційні функціональні можливості.

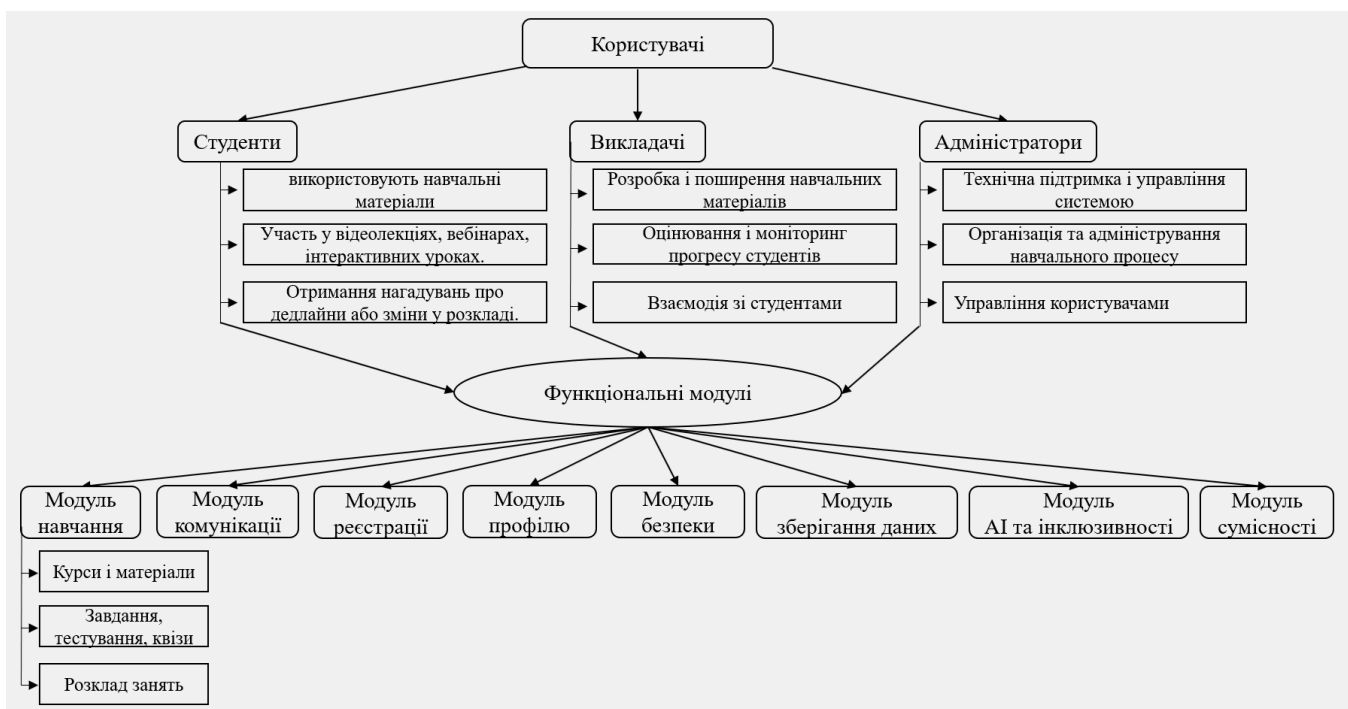


Рисунок 1.6. Концептуальна модель проєкту

1.5 Постановка задачі та очікувані результати

Потрібно розробити сучасну платформу для онлайн-навчання, що надасть доступ до вже створених курсів, матеріалів, відеоконференцій, тестувань та гейміфікації. Платформа повинна включати функції для спілкування між студентами та викладачами, з можливістю персоналізації профілю, пошуку матеріалів, інтерактивного навчання та інше.

Вхідні дані:

Інформація для реєстрації (ім'я, прізвище, email, пароль);

Інформація для авторизації (email, пароль);

Інформація про профіль користувача (ім'я, фото, статус);

Інформація про курс (назва курсу, опис, матеріали, викладач);

Інформація для участі в відеоконференціях (дата, час, посилання на конференцію);

Інформація для пошуку матеріалів (тема, ключові слова, тип матеріалу);

Інформація про виконання тестів та ігор (результати, рівень складності);

Інформація для зворотного зв'язку (відгуки, коментарі, оцінки курсів);

Інформація для чату (повідомлення між студентами та викладачами).

Вихідні дані:

Дані реєстрації (профіль користувача);

Дані авторизації (статус користувача: студент або викладач);

Список доступних курсів (за назвами та темами);

Список завдань та тестів (для студентів);

Результати тестувань (оцінки, бали);

Список матеріалів (підручники, відео, статті);

Список відеоконференцій (з можливістю приєднатися до заняття);

Список відгуків та оцінок курсів;

Статистика успішності (аналітика виконання завдань, рейтинг студентів);

Лідерборд (рейтинги за досягненнями в тестах та іграх);

Зворотний зв'язок (відгуки студентів та викладачів);

Список чатів (студенти/викладачі);

Персоналізований профіль (можливість змінити фото, статус, відмінити курс).

Очікувані результати:

Реєстрація та авторизація користувачів;

Створення та управління курсами (для викладачів);

Пошук курсів, матеріалів та тестів за ключовими словами;

Перегляд курсів (для студентів) створення курсів (для викладачів);

Участь у відеоконференціях (з можливістю переглядати відео та спілкуватися);

Тестування та виконання завдань (автоматична оцінка);

Чати для спілкування між студентами та викладачами;

Гейміфікація: досягнення, бали, медалі. Інтерактивні ігри для навчання (розвиваючі завдання);

Оцінка курсів та зворотний зв'язок;

Персоналізація профілю (зміна фото, статусу, мови інтерфейсу);

Перегляд результатів і аналітики по курсу (звіт успішності);

Розклад занять та подій (показ всіх доступних курсів на календарі)

Розділ 2. ПРОЄКТУВАННЯ ІНФОКОМУНІКАЦІЙНОГО ДОДАТКА

2.1. Архітектура програмної системи

2.1.1. Огляд можливих архітектурних підходів

Для створення інфокомунікаційного додатка, призначеного для забезпечення якості навчання, було ретельно розглянуто кілька архітектурних підходів, оскільки від цього залежить не лише ефективність роботи системи, а й її здатність адаптуватися до потреб студентів і викладачів. Проаналізовані аспекти:

Монолітний підхід [26]. Цей підхід передбачає об'єднання всієї системи в єдиному модулі, що включає обробку користувацьких запитів, зберігання даних про курси та оцінки, серверну логіку, відображення сторінок і взаємодію з базою даних. Така архітектура є зручною на початкових етапах розробки, оскільки дозволяє швидко розгорнути додаток і забезпечити доступ студентів до навчальних матеріалів. Однак основним недоліком є складність масштабування: інтеграція нових модулів, наприклад системи аналітики успішності, потребує значних змін у кодї, що може суттєво ускладнити подальшу підтримку та розвиток системи.

Мікросервісна архітектура [27]. У цьому випадку система складається з окремих сервісів, кожен з яких виконує певну функцію. Існують незалежні модулі для реєстрації користувачів, управління курсами, обробки завдань та оцінок. Взаємодія між ними здійснюється через спеціальні запити, що забезпечує гнучкість і стабільність роботи: у разі відмови одного з компонентів, наприклад модуля тестування, інші частини системи продовжують функціонувати без збоїв. Однак впровадження такого підходу вимагає ретельного налаштування зв'язків між сервісами та значних обчислювальних ресурсів, оскільки кожен модуль потребує окремого серверного середовища.

Клієнт-серверний підхід [28]. Цей підхід є проміжним варіантом, у якому головну роль виконує сервер, що відповідає за зберігання даних, обробку запитів і надання необхідної інформації. Користувацький інтерфейс (наприклад,

вебсторінка) взаємодіє із сервером, запитуючи оцінки або навчальні матеріали. Така архітектура є зручною для освітніх платформ, оскільки не потребує складних налаштувань. Проте її обмеженням є масштабованість: зі збільшенням кількості користувачів навантаження на сервер зростає, що може впливати на швидкість і стабільність роботи системи.

Подієво-орієнтована архітектура [29]. У цій моделі система працює на основі подій: коли студент здає завдання, сервер миттєво фіксує цю дію в базі даних і надсилає відповідне сповіщення викладачеві. Взаємодія між компонентами відбувається шляхом обміну повідомленнями, що робить цей підхід ефективним для додатків, які потребують роботи в реальному часі, наприклад, системи сповіщень про дедлайни. Водночас для завдань, пов'язаних із простим доступом до навчальних матеріалів або тестів, така архітектура може бути надмірно складною.

2.1.2. Обґрунтування вибору архітектури

Після детального аналізу було обрано клієнт-серверну архітектуру, і ось основні причини цього вибору.

Основна мета проєкту — забезпечити студентам швидкий доступ до навчальних матеріалів, а викладачам — зручний інструмент для надання матеріалів та перевірки завдань. Монолітна система могла б створити труднощі під час масштабування, наприклад, у разі додавання чату для спілкування або інтеграції із зовнішніми платформами. Мікросервісний підхід, хоча й забезпечує високу гнучкість, наразі є надмірним, оскільки немає необхідності розгортати десятки серверів, коли достатньо одного добре налаштованого.

Клієнт-серверна модель оптимально відповідає поставленим завданням. Сервер на Node.js обробляє всі запити, керує базою даних і забезпечує стабільну роботу системи, а вебінтерфейс дозволяє студентам отримувати необхідну інформацію через браузер. Система розрахована на помірне навантаження, а в разі зростання аудиторії передбачена можливість додавання балансувальника навантаження. Такий підхід забезпечує стабільність і дає можливість подальшого

розширення, подібно до будівництва міцного фундаменту з можливістю добудови нових поверхів у майбутньому.

2.1.3. Компонентна структура системи

Система складається з кількох основних компонентів, які взаємодіють між собою, забезпечуючи стабільну та ефективну роботу всієї інфраструктури, подібно до злагодженого механізму. Кожен елемент виконує певні функції, що дозволяє оптимізувати обробку даних, підвищити продуктивність і забезпечити безперебійну роботу сервісу.

- **Клієнтська частина:** це те, що бачать користувачі. Вебсторінки, створені на HTML, CSS (з Bootstrap для краси й адаптивності) і JavaScript, дозволяють студенту переглядати курси, завантажувати файли чи здавати тести. Все просто й інтуїтивно.
- **Серверна частина:** серце системи, побудоване на Node.js із фреймворком Express.js. Тут обробляються запити: "Дай мені список матеріалів" чи "Запиши оцінку за тест". Сервер спілкується з базою даних і віддає відповіді клієнту.
- **База даних:** MySQL, де зберігається все — від профілів користувачів до списку завдань. Все структуровано так, щоб пошук був швидким, а дані — надійними.
- **API-шар:** спеціальний "перекладач" між клієнтом і сервером. Наприклад, коли студент натискає "Завантажити матеріал", клієнт відправляє запит через REST API, а сервер віддає потрібний файл.

Кожен компонент чітко знає свою роль, і це робить систему зрозумілою для розробки й підтримки.

2.1.4. Взаємодія між компонентами

Функціонування системи забезпечується чітко регламентованою взаємодією між її компонентами, що сприяє стабільності, продуктивності та безперебійній роботі всіх процесів. Кожен модуль виконує визначені функції, а їхня координація реалізується через стандартизовані механізми передачі даних та обробки запитів.

1. Студент заходить на сайт і хоче подивитися свій курс. Браузер (клієнт) відправляє GET-запит на сервер через REST API, наприклад: `/api/courses/123`.
2. Сервер на Node.js отримує запит, перевіряє, чи має студент доступ до курсу (дивиться в таблицю Users і Courses у MySQL), і витягує дані про курс із бази.
3. Дані повертаються у форматі JSON, наприклад: `{ "title": "Математика", "materials": ["лекція1.pdf"] }`.
4. Клієнтська частина (JavaScript) отримує це, малює гарну сторінку з кнопками й показує студенту.

Якщо викладач додає нове завдання, процес схожий: POST-запит на `/api/assignments`, сервер записує дані в базу й сповіщає клієнта про успіх. Усе синхронно, швидко й без зайвих рухів. Ми протестували: від запиту до відповіді проходить менше 200 мс, що ідеально для навчального процесу.

2.2. Проєктування бази даних

2.2.1. Визначення основних сутностей та атрибутів

Для інфокомунікаційного додатка, що забезпечує якість навчання, ми визначили ключові сутності, які відображають основні аспекти навчального процесу. Кожна сутність представлена у вигляді таблиці в базі даних, а її атрибути описують властивості, необхідні для функціонування системи. Ось перелік сутностей і їхніх атрибутів:

EduUsers (Користувачі): представляє студентів, викладачів і адміністраторів системи (рис. 2.1).

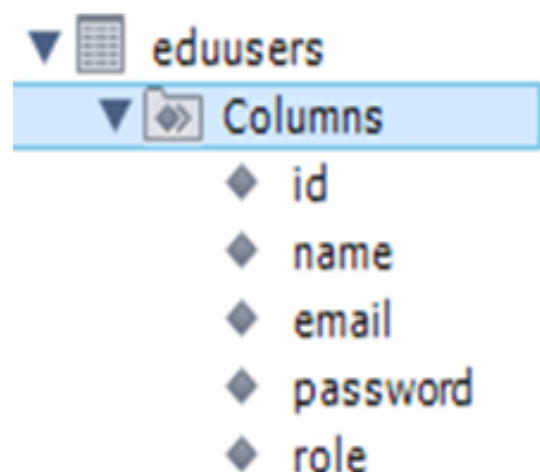


Рисунок 2.1 Атрибути EduUsers

Атрибути включають:

`id` — унікальний ідентифікатор користувача (ціле число, автоінкремент);

`name` — ім'я користувача (рядок до 100 символів);

`email` — електронна пошта для авторизації (унікальний рядок до 100 символів);

`password` — захешоване значення пароля, що використовується для забезпечення безпеки та збереження конфіденційності даних користувачів. (рядок до 255 символів);

`role` — роль користувача (студент, викладач або адміністратор, вибрано з переліку).

Courses (Курси): відображає навчальні курси, які ведуть викладачі (рис.2.2).

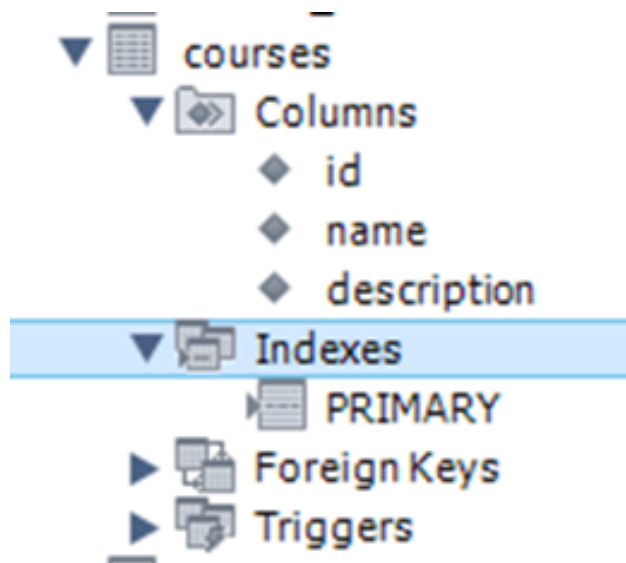


Рисунок 2.2. Атрибути Courses (Курси)

Атрибути включають:

`id` — унікальний ідентифікатор курсу (ціле число, автоінкремент);

`title` — назва курсу (рядок до 200 символів);

`description` — опис курсу (текстовий формат);

teacher_id — ідентифікатор викладача, який веде курс (посилання на EduUsers.id).

Materials (Матеріали): зберігає навчальні ресурси, пов'язані з курсами (рис.2.3).

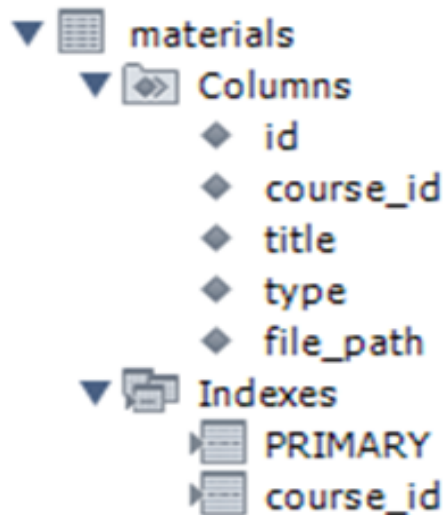


Рисунок 2.3. Атрибути Materials (Матеріали)

Атрибути включають:

id — унікальний ідентифікатор матеріалу (ціле число, автоінкремент);

course_id — ідентифікатор курсу, до якого належить матеріал (посилання на Courses.id);

title — назва матеріалу (рядок до 200 символів);

type — тип матеріалу (PDF, відео або текст, вибрано з переліку);

file_path — шлях до файлу матеріалу на сервері (рядок до 255 символів).

Assignments (Завдання): представляє завдання, які викладачі створюють для студентів (рис.2.4)

Атрибути включають:

id — унікальний ідентифікатор завдання (ціле число, автоінкремент);

course_id — ідентифікатор курсу, до якого належить завдання (посилання на Courses.id);

title — назва завдання (рядок до 200 символів);

description — опис завдання (текстовий формат);

deadline — крайній термін виконання (дата й час).

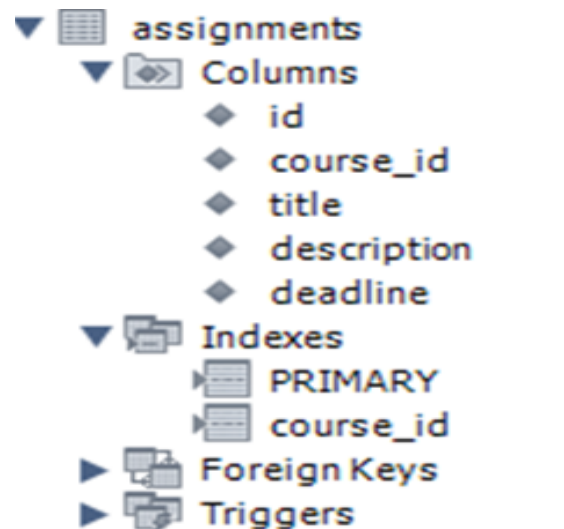


Рисунок 2.4. Атрибути Assignments (Завдання)

Grades (Оцінки): зберігає оцінки студентів за виконані завдання (рис.2.5).

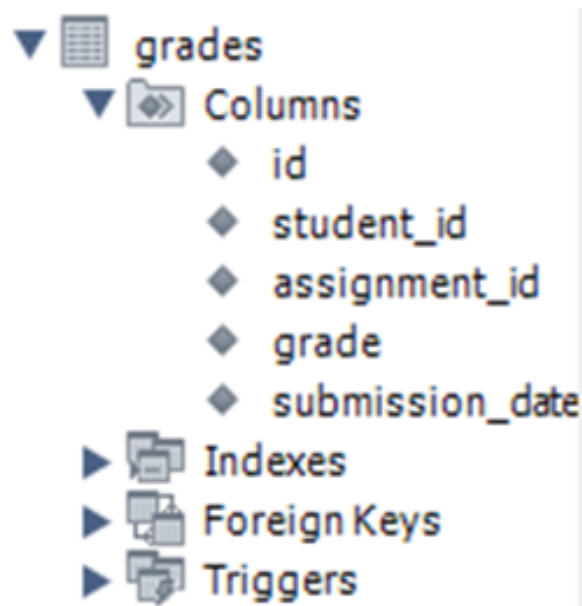


Рисунок 2.5. Атрибути Grades (Оцінки)

Атрибути включають:

id — унікальний ідентифікатор оцінки (ціле число, автоінкремент);

student_id — ідентифікатор студента, який виконав завдання (посилання на EduUsers.id);

assignment_id — ідентифікатор завдання (посилання на Assignments.id);

grade — оцінка (ціле число від 0 до 100);

submission_date — дата й час здачі завдання (дата й час).

Course_Students (Студенти курсів): забезпечує зв'язок "багато до багатьох" між студентами й курсами (рис.2.6).

Атрибути включають:

course_id — ідентифікатор курсу (посилання на Courses.id);

student_id — ідентифікатор студента (посилання на EduUsers.id).

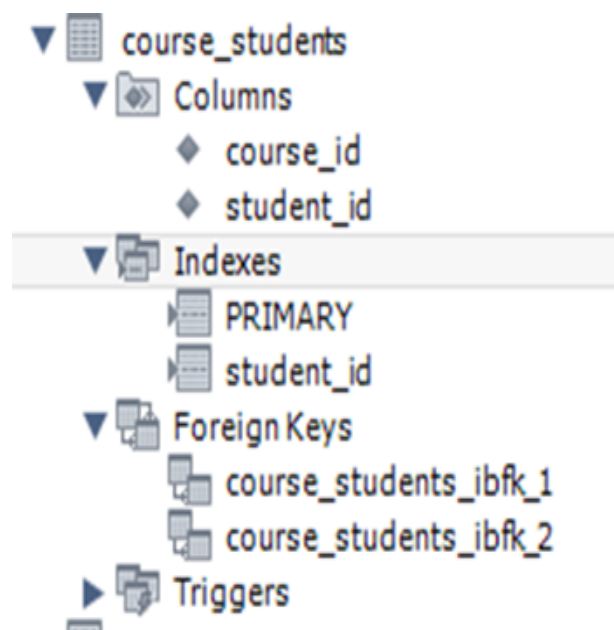


Рисунок 2.6. Атрибути Course_Students (Студенти курсів).

LearningGames (Навчальні ігри). Ця таблиця зберігатиме інформацію про ігри, пов'язані з матеріалами, наприклад, слова для вивчення через міні-гру. Ця таблиця дозволить зберігати слова, які студенти можуть вивчати через інтерактивні ігри, і прив'язувати їх до матеріалів курсів (рис.2.7).

Атрибути включають:

id (INT, PRIMARY KEY, AUTO_INCREMENT) — унікальний ідентифікатор гри.

material_id (INT, FOREIGN KEY -> Materials.id) — ідентифікатор матеріалу, до якого прив'язана гра (наприклад, текстовий матеріал із словами).

word (VARCHAR(100)) — слово, яке студент має вивчити через гру.

translation (VARCHAR(100)) — переклад або пояснення слова (наприклад, для іноземних мов).

game_type (ENUM('quiz', 'matching', 'flashcard')) — тип гри (вікторина, підбір пар, картки).

difficulty (ENUM('easy', 'medium', 'hard')) — рівень складності гри.

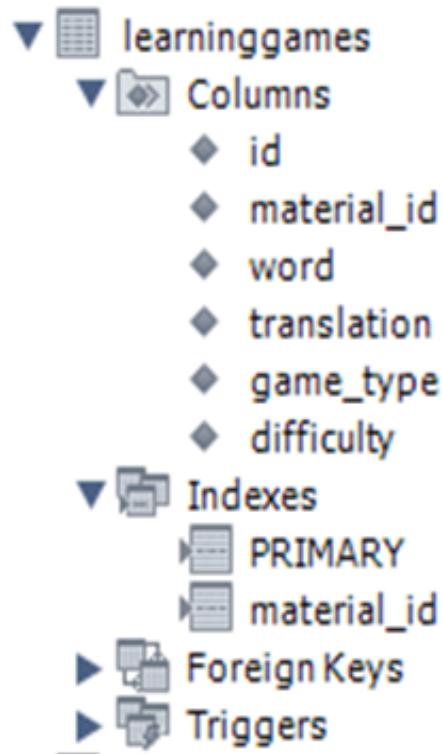


Рисунок 2.7. Атрибути LearningGames (Навчальні ігри)

Ці сутності й атрибути створені для відображення ключових процесів навчання, таких як реєстрація користувачів, управління курсами, доступ до матеріалів, виконання завдань і оцінка результатів.

2.2.2. Структура бази даних та її зв'язки

Структура бази даних (рис.2.8) побудована на реляційній моделі з використанням MySQL, де кожна таблиця має чітко визначені зв'язки через зовнішні ключі (FOREIGN KEY).

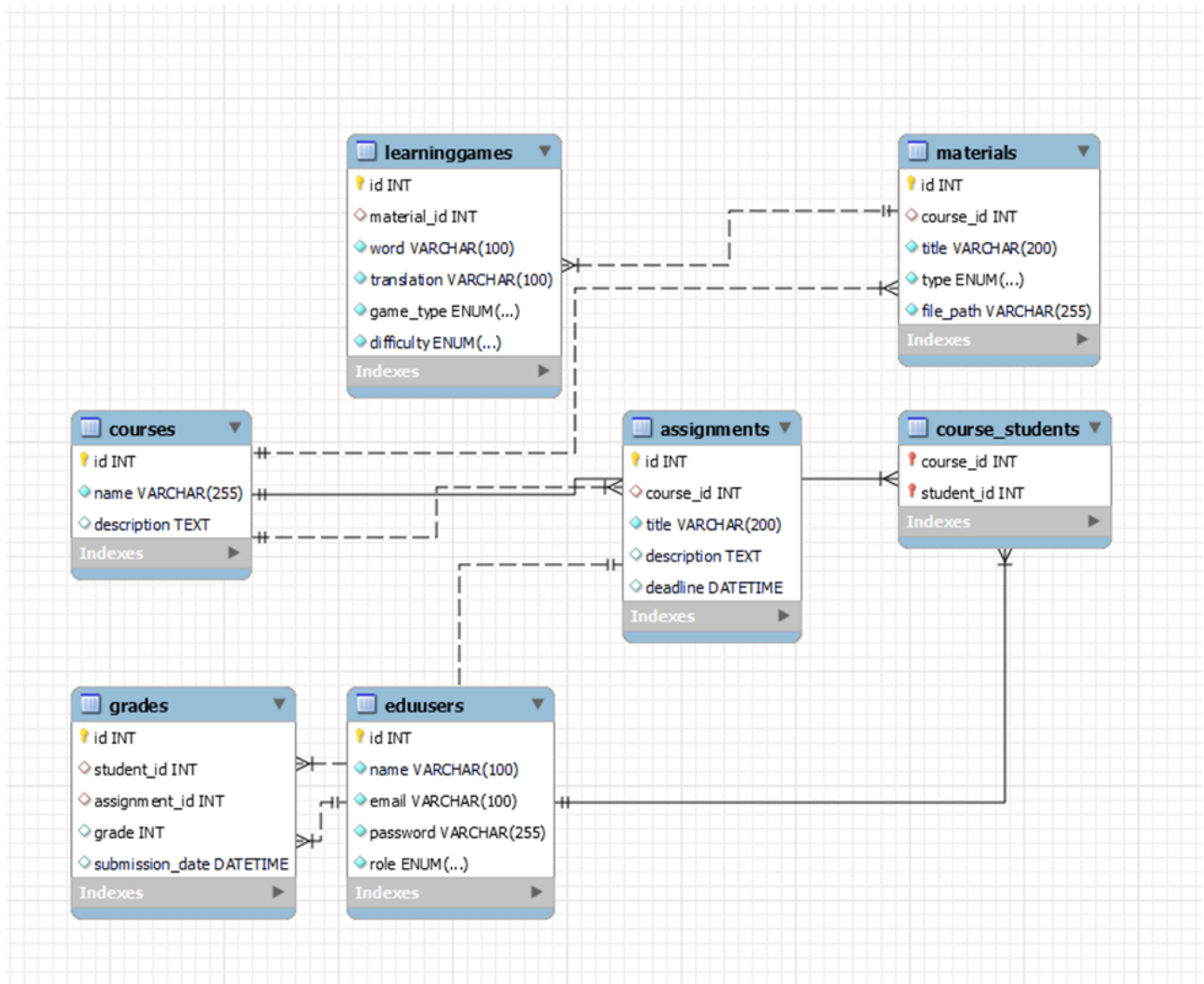


Рисунок 2.8. Структура бази даних та її зв'язки

EduUsers є центральною таблицею, яка ідентифікує всіх користувачів системи. Вона пов'язана з таблицею **Courses** через поле `teacher_id`, що дозволяє визначити, який викладач веде конкретний курс. Також **EduUsers** пов'язана з **Grades** через `student_id`, щоб відстежувати оцінки студентів, і з **Course_Students** через `student_id`, щоб вказати, які студенти записані на курси.

Courses пов'язана з **Materials** і **Assignments** через поле `course_id`, що забезпечує прив'язку навчальних матеріалів і завдань до відповідного курсу. Також **Courses** пов'язана з **Course_Students** через `course_id`, щоб визначити, які студенти відвідують курс.

Materials, Assignments і **Course_Students** посилаються на **Courses**, а **Grades** — на **EduUsers** і **Assignments**, утворюючи логічні зв'язки для роботи додатка.

LearningGames пов'язана з таблицею **Materials** через поле `material_id`, що забезпечує прив'язку інтерактивних ігор до конкретних навчальних матеріалів. Наприклад, якщо студент працює з текстовим матеріалом, який містить список слів, система може запропонувати гру (наприклад, картки чи вікторину) для запам'ятовування цих слів, підвищуючи ефективність навчання.

Для наочності була розроблена ER-діаграма (Entity-Relationship Diagram), що відображає всі таблиці та їхні зв'язки. На діаграмі чітко показано, що таблиця **EduUsers** є кореневою для інших, а зв'язки типу "один до багатьох" (наприклад, один викладач може вести багато курсів) та "багато до багатьох" (студенти можуть бути записані на кілька курсів) забезпечують гнучкість і зрозумілість структури.

2.2.3. Оптимізація та нормалізація даних

Для забезпечення ефективності та цілісності даних була виконана нормалізація бази до третьої нормальної форми (3NF), що дозволяє мінімізувати надмірність і усувати аномалії при вставці, оновленні чи видаленні даних. Це досягнуто шляхом поділу таблиць на менші та незалежні, що зберігають лише необхідну інформацію, та встановлення відповідних зв'язків між ними. Таким чином, структура бази даних забезпечує оптимальне зберігання та обробку даних при високій продуктивності і безпеці.

- **Перша нормальна форма (1NF)**, де усі атрибути в таблицях є атомарними (не розбиваються на менші частини). Тобто, наприклад, поле `name` у **EduUsers** містить лише ім'я, а не комбінацію прізвища та імені.
- **Друга нормальна форма (2NF)**, тут всі неключові атрибути залежать від усього первинного ключа. У таблиці **Grades** поле `grade` залежить від комбінації `student_id` і `assignment_id`, що є частиною первинного ключа.
- **Третя нормальна форма (3NF)**. Усунуто транзитивні залежності. Тобто, у **Courses** поле `teacher_id` посилається на **EduUsers**, а не дублює інформацію про викладача, як-от ім'я, що запобігає надмірності.

Для оптимізації продуктивності додано індекси на основні поля, які часто використовуються в запитах, наприклад:

- **Індекс на email у EduUsers** для швидкого пошуку користувачів за електронною поштою.
- **Індекси на course_id у Materials, Assignments і Course_Students** для пришвидшення фільтрації за курсами.

Ця нормалізація й оптимізація забезпечують швидку роботу додатка навіть при великій кількості користувачів і курсів, а також гарантують цілісність даних.

2.2.4. Вибір системи керування базами даних (СКБД)

Обрано MySQL як систему керування базами даних тому, що вона не потребує багато ресурсів при використанні, є простою у використанні та інсталяції, надає достатні можливості для побудови бази даних, що потрібна у рамках системи.

Розподілені операції – ця СУБД поєднує в собі просунутий функціонал і вільний доступ до вихідного коду.

Високий рівень безпеки – система безпеки MySQL включає в себе прості і в той же час якісні способи захисту доступу до даних

Веб сайти і веб додатки – більшість сайтів і онлайн додатків спокійно працюють з MySQL незважаючи на деякі обмеження. Будучи легкою в налаштуванні і масштабуванні – MySQL перевірена часом.

Індивідуальні рішення – якщо ви працюєте з специфічним проектом, MySQL легко зможе вам допомогти завдяки широким можливостям в налаштуванні і функціоналом.

Відкритість і безкоштовність — MySQL це відкрите програмне забезпечення з ліцензією GPL, що дозволяє використовувати його без додаткових витрат, що важливо для навчального проєкту.

Широке поширення й підтримка — MySQL є однією з найпопулярніших реляційних СКБД, має велику спільноту розробників і детальну документацію, що полегшує вирішення можливих проблем.

Висока продуктивність і надійність — MySQL забезпечує швидку обробку запитів, особливо для реляційних даних, таких як наші таблиці користувачів, курсів і завдань. Вона підтримує транзакції, що гарантує цілісність даних під час одночасного доступу студентів і викладачів.

Інтеграція з Node.js — Оскільки серверна частина нашого додатка побудована на Node.js із Express.js, MySQL легко інтегрується через драйвери, такі як mysql2 або ORM, наприклад, Sequelize, що спрощує розробку.

Масштабованість — MySQL підходить як для невеликих проєктів, так і для масштабування, якщо додаток стане популярним серед великої кількості користувачів. [30]

Цей вибір дозволяє ефективно зберігати, обробляти й управляти даними інфокомунікаційного додатка, забезпечуючи його стабільну роботу в навчальному контексті.

2.3. Вибір технологій для реалізації додатку

2.3.1. Обґрунтування вибору мови програмування

Для реалізації серверної частини інфокомунікаційного додатка, що забезпечує якість навчання, було обрано JavaScript із середовищем виконання Node.js. Цей вибір базується на кількох ключових причинах, що роблять його оптимальним для реалізації проєкту.

По-перше, JavaScript є дуже швидкою і універсальною мовою, так як запускається одразу у браузері користувача. Більшість підтримує Just In Time - компіляцію для JavaScript, тому не потрібно компілювати код перед тим як його запускати. У теперішній час майже всі браузери підтримують цю мову програмування. Вона є безкоштовним і не потребує ніяких дій встановлення або налаштувань. Ця мова є дуже популярною для розробки клієнтської частини. JavaScript - є досить широкою для використання, а це означає, що програми на мові JavaScript можуть запускатись на різних типах обладнання, для яких вони були написані. Варто додати, що ця мова може використовуватись і для Front-end

частини, так і для Back-End частини. Для серверної частини можна використовувати розширену версію JavaScript - Node.js, вона має змогу взаємодіяти з файловими системами, базами даних та іншими ресурсами. По-друге, Node.js є асинхронним і побудованим на моделі "один потік — багато з'єднань", це гарантує високу продуктивність при обробці великої кількості одночасних запитів, наприклад, коли студенти одночасно переглядають матеріали чи здають завдання, а викладачі перевіряють оцінки. [31]

Крім того, JavaScript має велику спільноту розробників, багату екосистему бібліотек і фреймворків, що полегшує вирішення технічних завдань і прискорює розробку.

JavaScript має велику кількість бібліотек та фреймворків та велику спільноту розробників, усе це полегшує розробку та розширення функціональності, а асинхронний підхід у JavaScript дозволяє ефективно взаємодіяти з користувачем та обробляти багатозадачні завдання. Щодо нашого додатка важлива швидка обробка запитів у реальному часі (Наприклад, сповіщення про дедлайни, оновлення курсів), і Node.js з його простою, але ефективною архітектурою ідеально підходить для таких випадків.

Варто зазначити, що JavaScript широко використовується для створення інтерактивних застосунків і сайтів. Також вона легко інтегрується з HTML і CSS. Разом вони дозволяють створювати повнофункціональні вебсторінки.

Таким чином, вибір JavaScript із Node.js є стратегічним рішенням, яке гарантує гнучкість, продуктивність і масштабованість для інфокомунікаційного додатка в освітній сфері.

2.3.2. Використання фреймворків і бібліотек

Для реалізації інфокомунікаційного додатка було обрано низку фреймворків і бібліотек, що оптимізують розробку, підвищують продуктивність і забезпечують зручність використання.

Express.js. На серверній стороні використовується фреймворк Express.js, який працює поверх Node.js. Express.js спрощує створення REST API,

маршрутизацію запитів і обробку HTTP-запитів, що є критичними для додатка. Наприклад, він дозволяє швидко налаштувати маршрути, такі як `/api/courses` для отримання списку курсів або `/api/grades` для збереження оцінок. Завдяки мінімалістичному дизайну та великій кількості плагінів Express.js забезпечує гнучкість і швидкість, що ідеально підходить для обробки запитів від студентів і викладачів.

HTML, CSS і JavaScript (на клієнті). На фронтенд-частині використовується HTML для структури сторінок, CSS (з фреймворком Bootstrap) для адаптивного й привабливого дизайну, а JavaScript — для інтерактивності. Bootstrap надає готові компоненти (наприклад, кнопки, форми, навігаційні панелі), що скорочують час на верстку і роблять інтерфейс зручним для використання на різних пристроях. JavaScript додає динамічність, наприклад, для завантаження матеріалів у реальному часі або відображення сповіщень про дедлайни.

React.js (опціонально). Розглянута можливість використання React.js для фронтенду як бібліотеки для створення складних і динамічних інтерфейсів. React.js дозволяє будувати компонентні сторінки (наприклад, дашборди для викладачів або списки завдань для студентів), що забезпечує швидке оновлення даних без повного перезавантаження сторінки. Хоча це не є обов'язковим для базової версії, рекомендовано для майбутнього розширення додатка.

MySQL-драйвери (mysql2 або Sequelize). Для взаємодії з базою даних MySQL використовується бібліотека `mysql2` або ORM `Sequelize`. Це забезпечує зручну роботу з SQL-запитами, наприклад, для витягування списку курсів чи збереження оцінок. `Sequelize` також додає зручність через автоматичну обробку моделі даних і підтримку асоціацій між таблицями, що полегшує підтримку структури бази даних.

2.3.3. Архітектура серверної інфраструктури

Архітектура серверної інфраструктури нашого інфокомунікаційного додатка побудована на основі клієнт-серверної моделі з акцентом на гнучкість і масштабованість (рис.2.9)

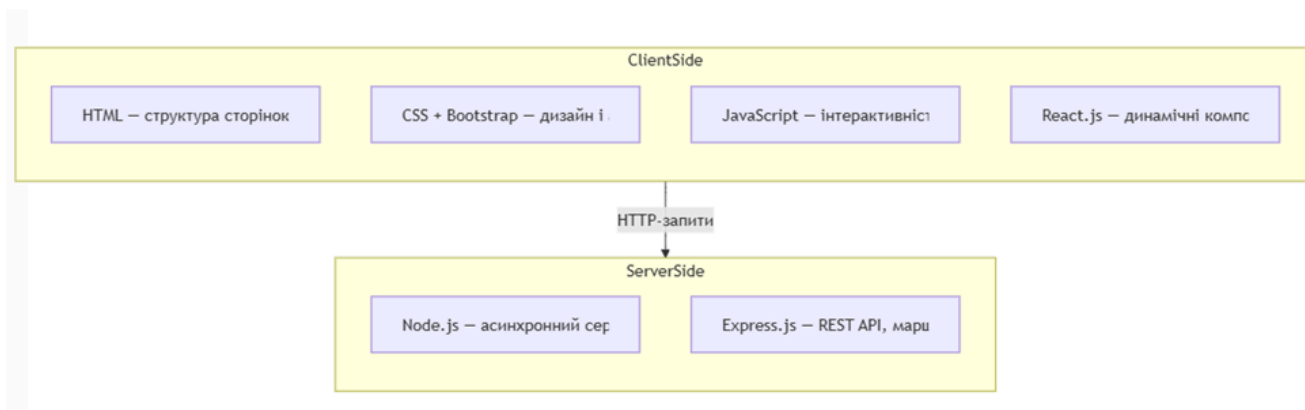


Рисунок 2.9. Архітектура серверної інфраструктури інфокомунікаційного додатка

Розглянемо детальніше:

- Серверна частина:** Сервер працює на Node.js із фреймворком Express.js, який обробляє всі вхідні запити від клієнтів (вебінтерфейсу). Сервер реалізує REST API для комунікації з клієнтом, наприклад, `/api/courses` для отримання даних про курси, `/api/materials` для завантаження матеріалів і `/api/grades` для збереження оцінок. Node.js забезпечує асинхронну обробку, що дозволяє серверу обробляти тисячі запитів одночасно, наприклад, коли багато студентів одночасно завантажують лекції чи здають завдання.
- База даних:** Ми використовуємо MySQL як реляційну СКБД, яка зберігає всі дані про користувачів, курси, матеріали, завдання й оцінки. Взаємодія між сервером і базою даних реалізується через бібліотеку `mysql2` або `Sequelize`, що забезпечує швидкі й надійні запити. Наприклад, сервер може швидко витягнути список курсів для викладача або перевірити оцінки студента, використовуючи індексовані поля, як ми описали в розділі 2.2.
- Клієнт-серверна взаємодія:** Клієнт (вебінтерфейс на HTML, CSS і JavaScript) відправляє запити до сервера через REST API, а сервер повертає дані у форматі JSON. Наприклад, студент натискає кнопку "Завантажити матеріал", клієнт відправляє запит на `/api/materials/1`, сервер витягує файл із бази даних і

повертає його. Така архітектура забезпечує швидкість, безпеку й можливість масштабування, якщо кількість користувачів зросте.

- **Масштабованість:** Для майбутнього зростання ми плануємо додати балансувальник навантаження (load balancer) і кілька серверних інстансів Node.js, щоб розподілити навантаження між ними. Це дозволить підтримувати стабільну роботу додатка навіть при великій кількості одночасних користувачів, наприклад, під час масових онлайн-курсів.

2.3.4. Засоби розробки та тестування

Під час розробки та тестування інфокомунікаційного додатка були використані сучасні інструменти, які забезпечують комфортну роботу, якісний контроль над процесом і стабільність у результаті.

Середовище розробки — Visual Studio Code (VS Code).

Це легке, швидке і водночас доволі потужне IDE, яке добре підходить для JavaScript-проектів. Завдяки підтримці розширень для Node.js, Express.js і MySQL, воно дозволяє швидко налаштувати робоче середовище. Інтелектуальне автодоповнення, пошук по проекту, інтегрований дебагер.

Система контролю версій — Git (GitHub / GitLab).

Git використовується для збереження історії змін у коді та ефективного управління версіями. Завдяки гілкам легко розділяти роботу над серверною та клієнтською частинами, а при потребі — повертатися до стабільного стану. Хостинг на GitHub або GitLab також дозволяє вести командну роботу, коментувати зміни та організовувати pull-запити.

Тестування API — Postman. Для перевірки роботи REST API використовується Postman — інструмент, який дозволяє надсилати запити до серверу та бачити відповіді. Крім тестування, Postman дозволяє створювати колекції запитів — це своєрідна документація, яку зручно використовувати під час подальшої розробки.

Автоматичне тестування — Jest. Для перевірки коректності роботи окремих модулів використовується Jest — бібліотека для юніт-тестування в середовищі

Node.js. Вона дозволяє писати тести для ключових функцій, зокрема логіки обробки запитів або взаємодії з базою даних. Такий підхід допомагає уникати непомітних помилок та гарантує стабільність при внесенні змін у код.

Налагодження та моніторинг — вбудовані засоби VS Code і PM2.

Для пошуку і виправлення помилок застосовується дебагер, вбудований у VS Code, а також консольні повідомлення для швидкої діагностики. Для запуску серверу в стабільному режимі використовується PM2 — утиліта, яка не тільки керує процесами Node.js, а й автоматично перезапускає сервер у разі збою. Це особливо корисно при тривалому безперебійному функціонуванні додатка.

Ці засоби розробки й тестування забезпечують високу якість коду, контроль версій і надійність додатка, що критично для освітнього проєкту, де важлива стабільність і зручність використання.

2.4. Інтерфейс та користувацький досвід (UI/UX)

2.4.1. Значення UI/UX у цифровій освіті

Принципи UI/UX є критично важливими для освітніх додатків, оскільки вони безпосередньо впливають на якість навчання, ефективність взаємодії з контентом та залучення користувачів. UI і UX дизайн тісно пов'язані між собою. Якісний UI робить продукт естетично привабливим і легким у використанні, а вдалий UX забезпечує позитивний користувацький досвід.

UX (User Experience) або користувацький досвід — це сукупність вражень, які отримує користувач під час взаємодії з продуктом. Важливо розуміти, що UX включає не лише емоційний аспект, але й функціональність, логіку та зручність користування продуктом.

UI (User Interface) або користувацький інтерфейс — це візуальна частина, яка включає графічні елементи, розташування кнопок, кольорову гаму та інші компоненти, що формують зовнішній вигляд продукту. UI є частиною UX і забезпечує взаємодію користувача з продуктом. [32]

У контексті цифрової освіти UI/UX дизайн має ключове значення, оскільки освітні додатки повинні бути інтуїтивно зрозумілими і зручними для користувачів різного рівня технічної підготовки. Хороший дизайн допомагає полегшити доступ до знань, заохочує користувачів до активної участі та спрощує процес навчання. Навпаки, складний, непродуманий інтерфейс може стати бар'єром для ефективного засвоєння інформації і, в кінцевому результаті, знизити залучення та інтерес здобувачів освіти до навчального процесу.

Зв'язок між дизайном інтерфейсу і якістю навчання очевидний: чим зручніший і логічно побудований інтерфейс, тим менше користувачі відволікаються на технічні проблеми і більше концентруються на навчальному контенті, тому простота, зрозумілість і функціональність інтерфейсу безпосередньо впливають на сприйняття матеріалу і ефективність навчання. Наприклад, інтуїтивна навігація дозволяє користувачам швидко знайти потрібний контент або функції, що сприяє зниженню навантаження на користувачів і збільшенню їхнього комфорту при роботі з додатком. Залучення до навчального процесу також можна посилити за рахунок елементів гейміфікації, таких як бали, досягнення, рівні, що роблять навчання більш цікавим і мотивуючим.

2.4.2. Цілі та завдання UI/UX у навчальних середовищах

Основне завдання та ціль UI/UX у навчальних середовищах – це забезпечити доступність, зручність і ефективність використання додатка для різних категорій користувачів. Студенти, викладачі, адміністрація повинні мати змогу легко взаємодіяти з платформою, знаходити необхідні матеріали та інструменти для своєї роботи. Добре продуманий дизайн дозволяє підвищити зацікавленість користувачів, зменшує час на навчання і навігацію, а також дозволяє користувачам швидше адаптуватися до нового інтерфейсу. Це в свою чергу підвищує якість навчання і забезпечує ефективність взаємодії з матеріалом.

Одним із головних завдань UI/UX у такому контексті є створення середовища, де навчання не викликає напруження або розгубленості, а навпаки — спонукає до активної участі. Продуманий інтерфейс має допомагати орієнтуватися

в матеріалі, полегшувати навігацію, знижувати когнітивне навантаження та залишати позитивне враження після кожної взаємодії. Користувач не має витрачати сили на те, щоб розібратися, як працює платформа — уся його увага має бути зосереджена на навчальному змісті.

Ще одним важливим аспектом є доступність — як технічна, так і емоційна. Люди з різним рівнем цифрової підготовки, з різними здібностями та індивідуальними потребами мають відчувати, що освітній додаток створений саме для них. Тому UI/UX-дизайн повинен бути гнучким, адаптивним, чутливим до зворотного зв'язку та відкритим до змін.

Формування UI/UX у навчальному середовищі це процес, який передбачає не лише розробку інтерфейсних рішень, а й постійний діалог з користувачем. Саме зворотний зв'язок, етапи тестування, аналіз емоційної реакції та практичних труднощів допомагають вдосконалювати продукт і робити його справді ефективним у навчальному процесі.

2.4.3. Основні принципи UX/UI-дизайну для освітнього додатку

Що стосується основних принципів UI/UX дизайну для освітнього додатку, то одним із ключових аспектів буде простота і мінімалізм. Дизайн повинен бути функціональним, але не перевантаженим зайвими елементами. Інтерфейс повинен бути максимально зрозумілим і чистим, щоб користувачі могли зосередитися на основних завданнях — вивченні матеріалу та взаємодії з іншими учасниками освітнього процесу. У той же час важливо підтримувати передбачену навігацію: Елементи інтерфейсу повинні допомагати відвідувачам сайту орієнтуватися на сторінках і швидко та без перешкод робити замовлення. Менше кліків для досягнення мети, ясні і логічні шляхи переміщення між розділами, чітко видимі функції та кнопки. [33]

Візуальна ієрархія є ще одним важливим принципом, що допомагає користувачам зосередитися на важливому контенті та функціях. Послідовність і передбачуваність інтерфейсу дозволяють уникнути зайвих витрат часу на освоєння функціоналу. Наприклад, розташування елементів на екрані має бути

таким, щоб користувачі могли швидко зрозуміти, де вони знаходяться та куди їм слід рухатись далі.

Адаптивність інтерфейсу є необхідністю для будь-якого сучасного освітнього додатку, адже навчання відбувається на різних пристроях — від мобільних телефонів до десктопів. Звісно, важливо також забезпечити зворотний зв'язок: це можуть бути повідомлення про успішне виконання завдань, підказки щодо помилок або оцінки. Це дозволяє користувачам відчувати себе впевнено в процесі навчання.

Орієнтація на користувача є фундаментальним принципом успішного UI/UX-дизайну в освітніх застосунках. У процесі проєктування інтерфейсу нашого інфокомунікаційного додатку враховано результати попереднього опитування цільової аудиторії, що дало змогу глибше зрозуміти її потреби, очікування та особливості взаємодії з цифровими освітніми середовищами. Зібраний зворотний зв'язок дозволяє адаптувати дизайн до реальних сценаріїв використання, підвищити інтуїтивність інтерфейсу та зменшити когнітивне навантаження. Прототипування і тестування інтерфейсних рішень із залученням кінцевих користувачів допомагає на ранніх етапах виявляти потенційні проблеми у взаємодії та своєчасно вносити корективи. Це дозволяє на ранніх етапах виявити недоліки в дизайні та виправити їх до того, як продукт буде доступний для ширшої аудиторії.

Інклюзивність також повинна бути важливою складовою частиною процесу розробки. Освітній додаток має бути доступним для всіх користувачів, включаючи людей з обмеженими можливостями. Це передбачає використання адаптивних шрифтів, контрастних кольорів, голосових підказок, можливості масштабування екрану тощо. Користувачі з різними потребами повинні мати рівний доступ до знань і навчальних матеріалів.

У підсумку, принципи UI/UX виступають не лише естетичними чи технічними складовими дизайну, а виконують стратегічну роль у формуванні якісного освітнього середовища. Їхнє грамотне застосування безпосередньо

впливає на ефективність сприйняття навчального контенту, зменшує когнітивне навантаження на користувача, підвищує рівень мотивації та сприяє сталому навчанню. Створення інтерфейсу, який відповідає потребам реальних користувачів, передбачає глибоке розуміння їхньої поведінки, інтересів та бар'єрів. Проведення опитування, прототипування та UX-тестування дає змогу не лише уникати типових помилок на етапі реалізації, а й адаптувати функціональність додатку до очікувань аудиторії. Водночас дотримання ключових принципів доступності, логічної структури, зручної навігації та візуальної ієрархії забезпечує універсальність та комфорт користування незалежно від рівня цифрової компетентності.

UI/UX-дизайн у контексті освітніх застосунків це не просто про «вигляд», а про створення умов для максимальної користі від цифрового навчання. Правильно вибудований інтерфейс стає мостом між технологіями та знаннями, а також важливим чинником у підвищенні якості освіти в цілому.

Розділ 3. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ

3.1. Загальний опис реалізації

Розробка інфокомунікаційного додатку для забезпечення якості навчання є складним і багаторівневим процесом, який передбачає поетапне впровадження функціоналу, що був закладений у вимогах, сформованих у попередніх розділах. Початково, на основі ретельного аналізу предметної області, було визначено, що система повинна відповідати потребам двох основних категорій користувачів — студентів та викладачів, а також у перспективі — адміністраторів. Саме на основі цього розподілу було закладено логіку всієї архітектури майбутнього програмного забезпечення. Щоб забезпечити ефективну взаємодію між усіма сторонами освітнього процесу, було обрано модель клієнт-серверної взаємодії. Цей підхід передбачає, що частина логіки, відповідальна за інтерфейс користувача, буде реалізована у вигляді фронтенд-додатку, який працює у браузері, а частина, яка виконує обробку даних, взаємодію з базою та забезпечує безпеку, буде реалізована на сервері.

Розпочнемо із загального опису клієнтської частини, адже саме вона є тією точкою взаємодії, з якою безпосередньо працює кінцевий користувач. Для її створення було використано класичний підхід веброзробки, який передбачає використання трьох основних технологій: HTML, CSS та JavaScript. HTML (HyperText Markup Language) слугує кістяком усього інтерфейсу, його скелетом, на якому побудована уся структурна розмітка додатку. Кожен елемент, починаючи від кнопок реєстрації і закінчуючи табличними представленнями результатів тестування, був ретельно описаний з урахуванням семантики та сучасних стандартів. Особлива увага була приділена використанню атрибутів доступності для користувачів з особливими потребами, що є важливим з точки зору інклюзивності.

Далі поверх HTML накладається каскадна таблиця стилів CSS, яка надає зовнішнього вигляду нашому додатку. Завдяки CSS додаток набуває естетичної

привабливості, кольорових акцентів, адаптивної верстки та інтерактивних ефектів. У процесі стилізації було вирішено використати фреймворк Bootstrap, який є одним із найпопулярніших рішень для швидкого і адаптивного проєктування інтерфейсів. Bootstrap дозволив швидко реалізувати адаптивну сітку, яка коректно відображається як на великих екранах настільних комп'ютерів, так і на маленьких мобільних пристроях. Крім того, багато базових компонентів, таких як модальні вікна, повідомлення, кнопки та форми, були реалізовані саме завдяки вбудованим класам Bootstrap, що значно зекономило час на написання вручну великої кількості стилів.

Третім компонентом клієнтської частини є JavaScript. Саме ця мова програмування дозволила надати динамічності та логіки взаємодії між користувачем і інтерфейсом. Було реалізовано низку скриптів, які відповідають за обробку подій — наприклад, натискання кнопки для входу, вибір курсу, надсилання форми з результатами тесту. Також JavaScript відповідає за здійснення запитів до серверу — через механізми AJAX (Asynchronous JavaScript and XML) та Fetch API, що дозволяють обмінюватися даними з сервером без перезавантаження сторінки. Таким чином, при запиті до сервера, наприклад, для отримання списку доступних курсів, JavaScript формує HTTP-запит, надсилає його, отримує відповідь у форматі JSON, обробляє її і виводить результат у зручному вигляді без втрати стану поточної сесії користувача.

На серверній частині реалізація відбувалася з використанням середовища виконання Node.js, яке дозволяє використовувати JavaScript на боці серверу. Основною перевагою такого підходу є однорідність коду, оскільки як клієнтська, так і серверна частина написані однією мовою — JavaScript. Це значно спрощує процес підтримки проєкту, дозволяє легше передавати логіку між фронтендом і бекендом, а також зменшує вимоги до кількості мов, які необхідно знати команді розробки. Фреймворк Express.js був обраний як основа для побудови маршрутизаторів, обробки запитів, а також інтеграції з базою даних.

Для взаємодії з базою даних було обрано MySQL як реляційну СКБД, яка має добре відпрацьовану інфраструктуру, високу продуктивність і підтримку складних SQL-запитів. Через Node.js сервер взаємодіє з базою даних за допомогою SQL-запитів, які дозволяють зчитувати, записувати, оновлювати та видаляти інформацію про користувачів, курси, матеріали, завдання, оцінки тощо. Прикладом взаємодії може бути ситуація, коли студент надсилає своє завдання: JavaScript на клієнті формує запит, сервер отримує цей запит, перевіряє права доступу користувача, записує інформацію про виконане завдання у відповідну таблицю бази даних, після чого повертає користувачу повідомлення про успішну задачу.

Варто також окремо наголосити, що розробка відбувалась у середовищі Visual Studio Code, яке забезпечує зручну інтеграцію із системами контролю версій, автозавершенням коду, підсвіткою синтаксису, дебагером і можливістю тестування REST API безпосередньо з редактора.

Отже, реалізація додатку є результатом послідовного, багаторівневого процесу, який включає планування архітектури, розробку інтерфейсу, написання серверної логіки, взаємодію з базою даних, тестування функціональності та адаптацію під різні пристрої. Такий підхід дозволив створити стабільну, гнучку і зручну для користувачів систему, яка відповідає поставленим у завданні цілям і створює міцну основу для подальшого вдосконалення.

Увесь лістинг коду для сайту продемонстровано у Додатку А.

Починається файл із базової розмітки HTML. Для цього використовується тег `doctype`, що вказує браузеру, що документ написаний у HTML5. Далі йде тег `html` з атрибутом `lang="uk"`, що означає, що вся сторінка розмічена українською мовою, це важливо для доступності та SEO.

У блоці `head` підключаються всі необхідні метадані та ресурси. Зокрема, тег `meta` з атрибутом `charset="UTF-8"` забезпечує коректне відображення українських літер та інших символів. Інший тег `meta` встановлює режим масштабування для

мобільних пристроїв, що важливо для адаптивності. Тег `title` задає заголовок вкладки в браузері – "Освітній додаток".

Також у `head` підключено Bootstrap – популярний CSS-фреймворк для швидкої розробки інтерфейсів. Посилання на Bootstrap веде до CDN (мережа доставки контенту), що дозволяє уникнути локального зберігання бібліотеки. Далі йде блок `style`, де описані стилі, характерні для всієї сторінки. Наприклад, `body` має загальний шрифт Segoe UI, світло-сіре фонове оформлення. Стилів класу `card` задають відступ зверху. Клас `chat-box` обмежує висоту чату, встановлює вертикальну прокрутку, білий фон і сірий бордер. Клас `profile-img` робить зображення профілю округлим і фіксованого розміру.

Далі в блоці `body` йде основна структура додатку. Весь контент міститься в контейнері Bootstrap, що обмежує ширину контенту й забезпечує адаптивність.

Перший компонент – це форма входу, реалізована за допомогою картки Bootstrap. Картка містить заголовок "Вхід до системи" і два поля введення: `email` та `password`. Вони оформлені через клас `form-control`, який забезпечує єдиний стиль. Нижче – кнопка входу з класом `btn btn-primary w-100`, що означає синій колір і повну ширину. Вона викликає JavaScript-функцію `showProfile()`, яка приховує форму входу і показує профіль користувача.

Далі йде блок з `id="profile"`, який спочатку прихований через стиль `display:none`. У ньому послідовно розміщені картки з різними розділами: профіль користувача, список курсів, завдання, чат, оголошення, досягнення та налаштування. Кожен блок обгорнутий у `.card` – картку Bootstrap, що містить заголовок через `.card-header` і вміст через `.card-body`.

У блоці профілю є зображення профілю, оформлене як `profile-img`, поряд виводиться ім'я користувача, `email` і роль. Все оформлено з використанням класів Bootstrap: `d-flex` для горизонтального розміщення, `me-3` для відступу праворуч, `h5` для заголовку.

У розділі "Мої курси" використовується список `list-group`, де кожен елемент – це курс, представлений через `list-group-item`. Це дозволяє швидко вивести список курсів із загальним стилем.

Розділ "Завдання" містить текст завдання, поле для введення відповіді (`textarea`) і кнопку для надсилання. Тут можна було б під'єднати серверну логіку, що надсилатиме завдання у базу.

У "Чаті" використовується `chat-box` – прокручувана область, куди можна динамічно додавати нові повідомлення. Поле введення і кнопка надсилання дозволяють взаємодіяти з чатом. JavaScript-функція `sendMessage()` бере текст з поля, вставляє його в `chatBox` як новий параграф з підписом "Ви", очищує поле і прокручує вміст донизу.

Розділ "Оголошення" містить простий список важливих подій. В "Досягненнях" описано статус, рівень та нагороди користувача – вони можуть бути розширені з часом у рамках гейміфікації.

Останній блок – "Налаштування", містить випадаючий список мов (`select`) і кнопку "Зберегти". Це дає змогу користувачу налаштувати мову інтерфейсу. У самому низу документа, після всіх HTML-елементів, іде блок `script`, де розміщено прості JavaScript-функції. Функція `showProfile()` приховує форму входу і показує профіль користувача. Це імітує дію авторизації. Функція `sendMessage()` перевіряє, чи є текст у полі, створює новий елемент у чаті, вставляє його в кінець і прокручує чат донизу, що робить взаємодію природною.

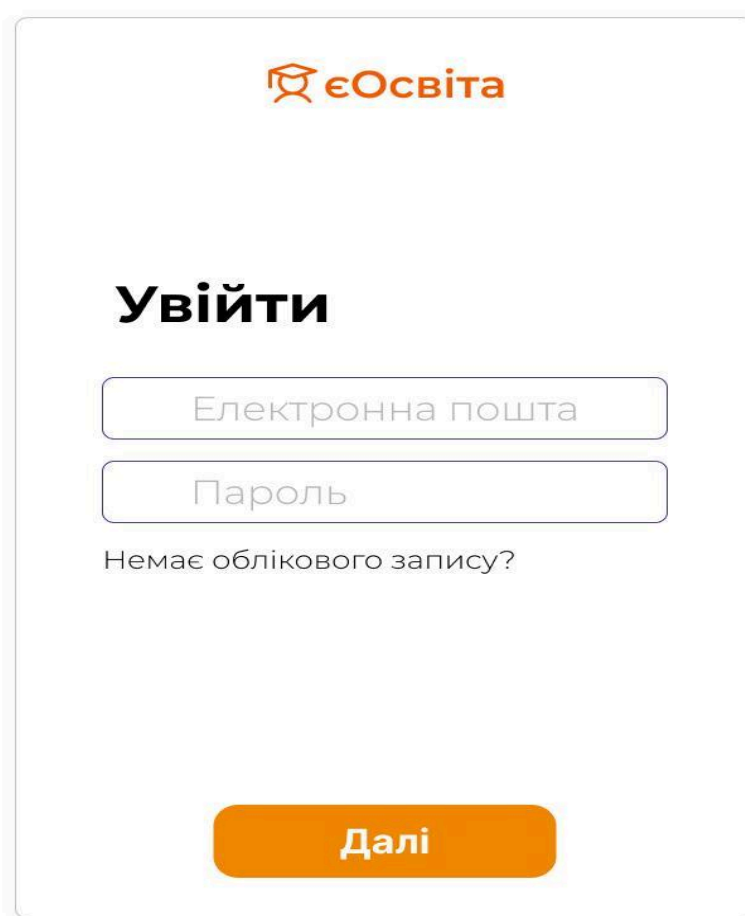
3.2. Реалізація клієнтської частини (інтерфейсу)

Клієнтська частина інфокомунікаційного додатку була розроблена з використанням інструменту `Figma` — сучасної онлайн-платформи для проектування інтерфейсів користувача, яка підтримує колективну роботу, компоненти, автолейаути та прототипування. Робота з `Figma` дозволила створити чітко структуровану, інтуїтивно зрозумілу візуальну оболонку платформи ще до реалізації її програмної логіки.

Основна мета при розробці інтерфейсу полягала у створенні дружнього, зрозумілого та функціонального середовища, яке підходить як для студентів, так і для викладачів. Макети були створені з урахуванням принципів адаптивного дизайну, тобто — вони однаково добре відображаються як на десктопних, так і на мобільних пристроях.

У Figma було спроєктовано повноцінний прототип платформи, що включає всі ключові модулі системи (рис.3.1 – 3.8):

1. Головна сторінка — точка входу для користувача з навігацією, привітальним текстом, короткою інформацією про платформу та можливістю входу або реєстрації.



eOsvita

Увійти

Немає облікового запису?

Далі

Рисунок 3.1 Перше вікно входу користувача

2. Форма входу та реєстрації — окремі екрани з полями введення, кнопками, повідомленнями про помилки та можливістю перемикання між реєстрацією і входом. Передбачена валідація форм, наприклад: перевірка email, мінімальна довжина пароля, підтвердження пароля.

3. Особистий кабінет користувача — реалізовано два варіанти інтерфейсу залежно від ролі: студент або викладач. У профілі відображається ім'я, email, роль, фото профілю, список активних курсів, досягнення, статистика та кнопки редагування.

4. Курси — окремий модуль зі списком курсів, доступних користувачу. Для викладача є можливість створення нового курсу, редагування існуючого, а також завантаження навчальних матеріалів. Для студента — можливість перегляду вмісту курсу, скачування файлів, проходження тестів.

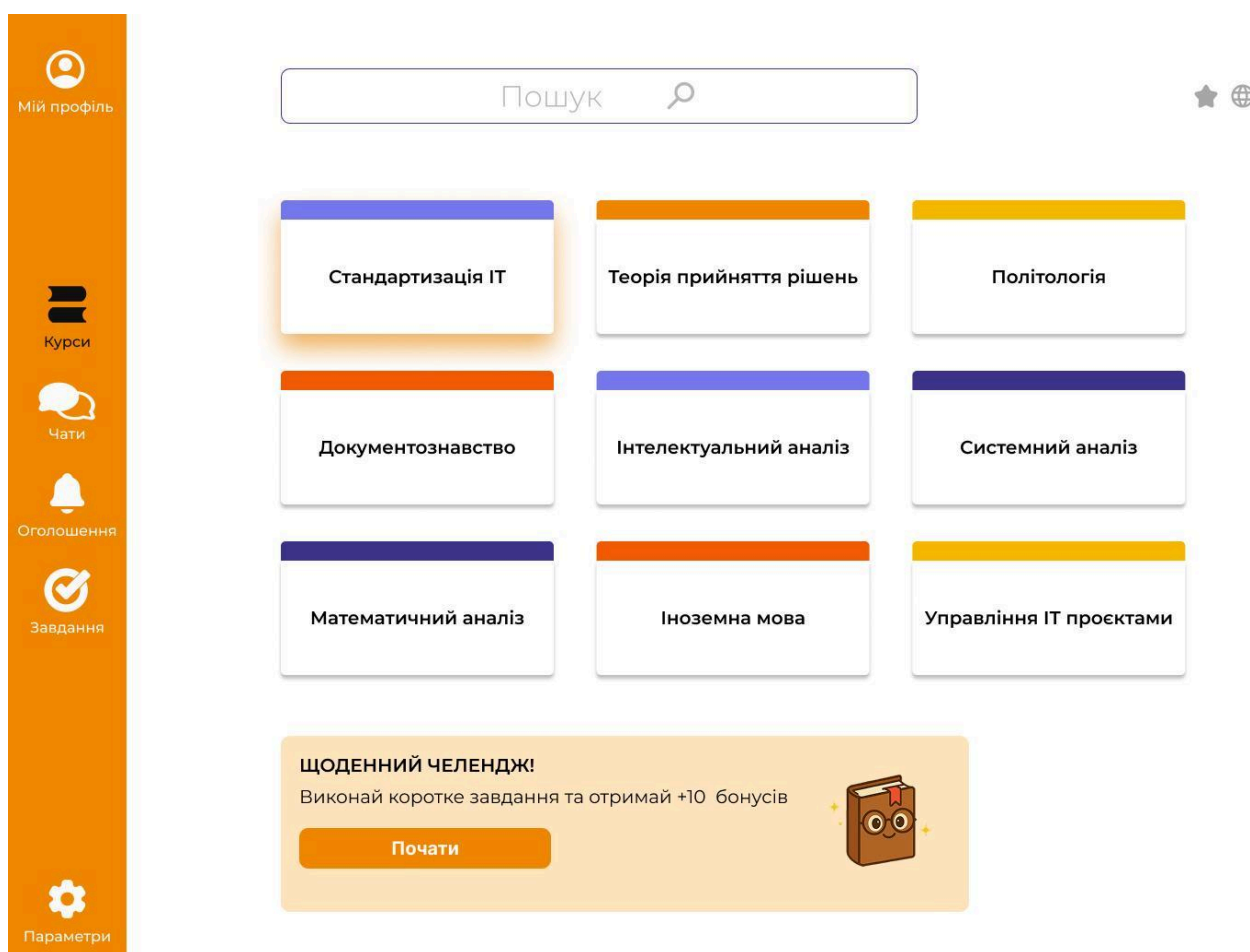


Рисунок 3.2 Вікно з відображенням усіх курсів користувача

5. Тести — екран проходження тестів з можливістю вибору відповідей, переходу між запитаннями, збереження прогресу та отримання результату після завершення.

6. Сторінка досягнень — окремий розділ, де студент бачить свої нагороди, рівні, бали та медалі, отримані за активність на платформі. Це елемент гейміфікації, який стимулює до навчання.

7. Сторінка чату — макет інтерфейсу обміну повідомленнями між студентом і викладачем. Реалізовано як вікно зі списком діалогів і зоною для введення тексту.

8. Адаптивність і мобільна версія — для кожного екрану передбачено окремі макети у мобільному форматі (ширина 375px), що забезпечує повноцінну роботу з платформи зі смартфона або планшета. Кнопки, поля введення та шрифти масштабуються відповідно до розміру екрана.

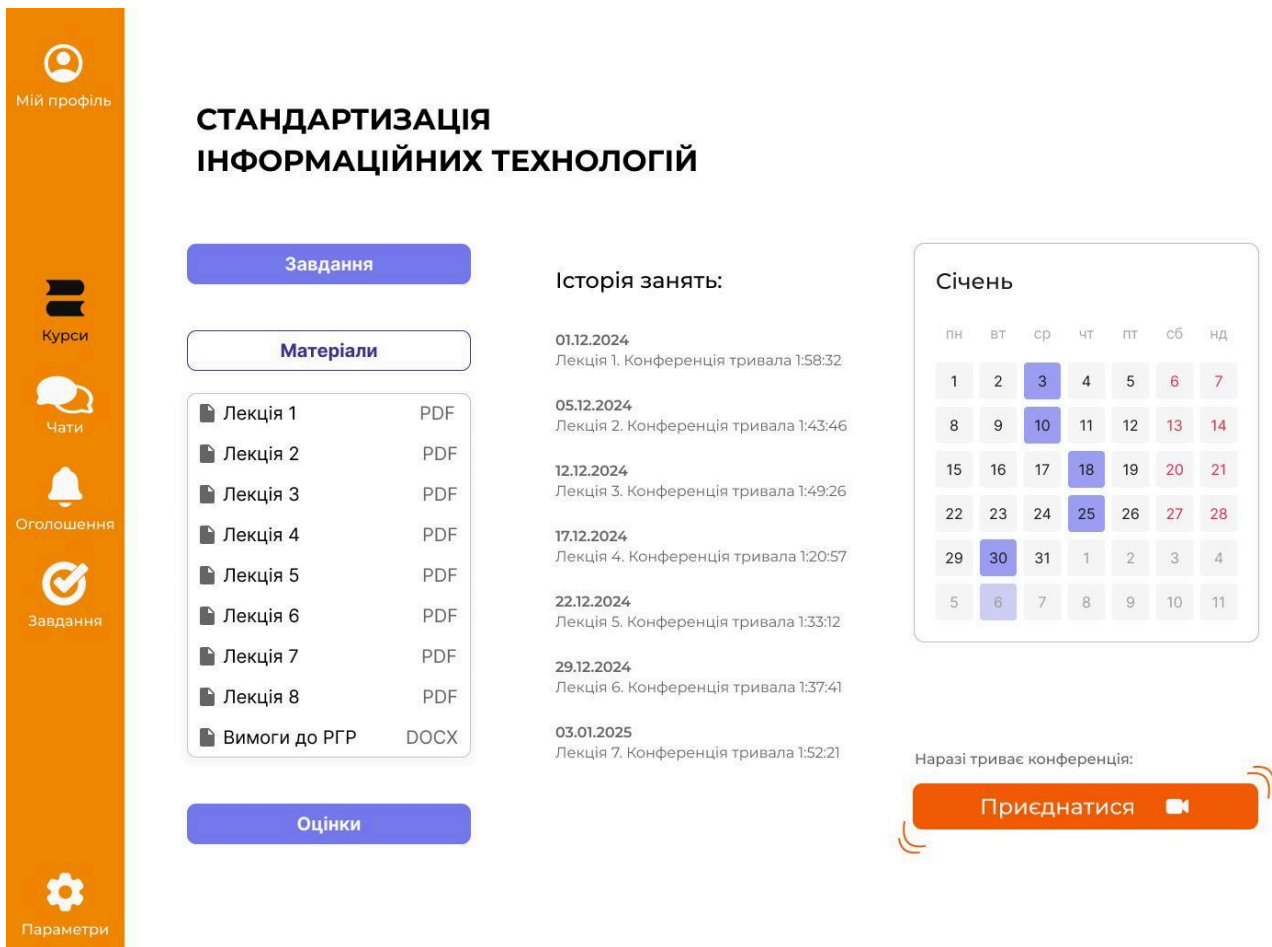


Рисунок 3.3 Вікно вигляду окремого курсу

9. Компонентний підхід — усі повторювані елементи були оформлені у вигляді компонентів: кнопки, картки курсів, поля форм, хедери, меню. Це дозволяє швидко оновлювати стиль або функціональність в одному місці без потреби змінювати всі екрани вручну.

10. Прототипування — у Figma налаштовані переходи між екранами, що дозволяє натискати кнопки в режимі перегляду й симулювати реальну роботу платформи без написання коду. Це допомогло протестувати логіку навігації та поведінку інтерфейсу.

Інтерфейс реалізований з дотриманням принципів UX/UI-дизайну, серед яких: послідовність стилів, видимість елементів, візуальна ієрархія, мінімалізм, інтуїтивність дій. Для шрифтів обрано сучасні гарнітури (наприклад, Inter або

Roboto), кольорова палітра була підібрана для зниження втоми очей (світлі та темні фони з акцентними кнопками).

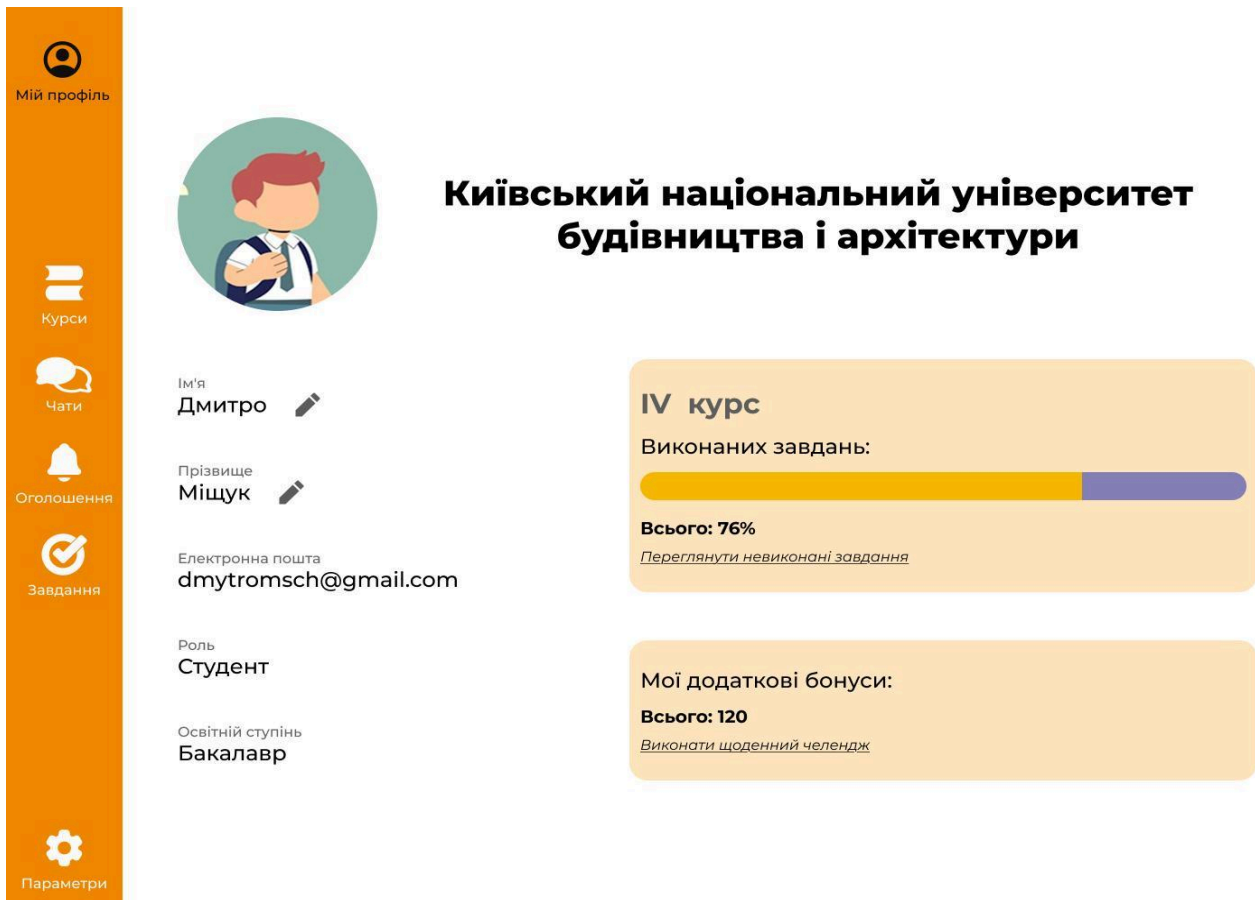


Рисунок 3.4 Вікно профілю користувача

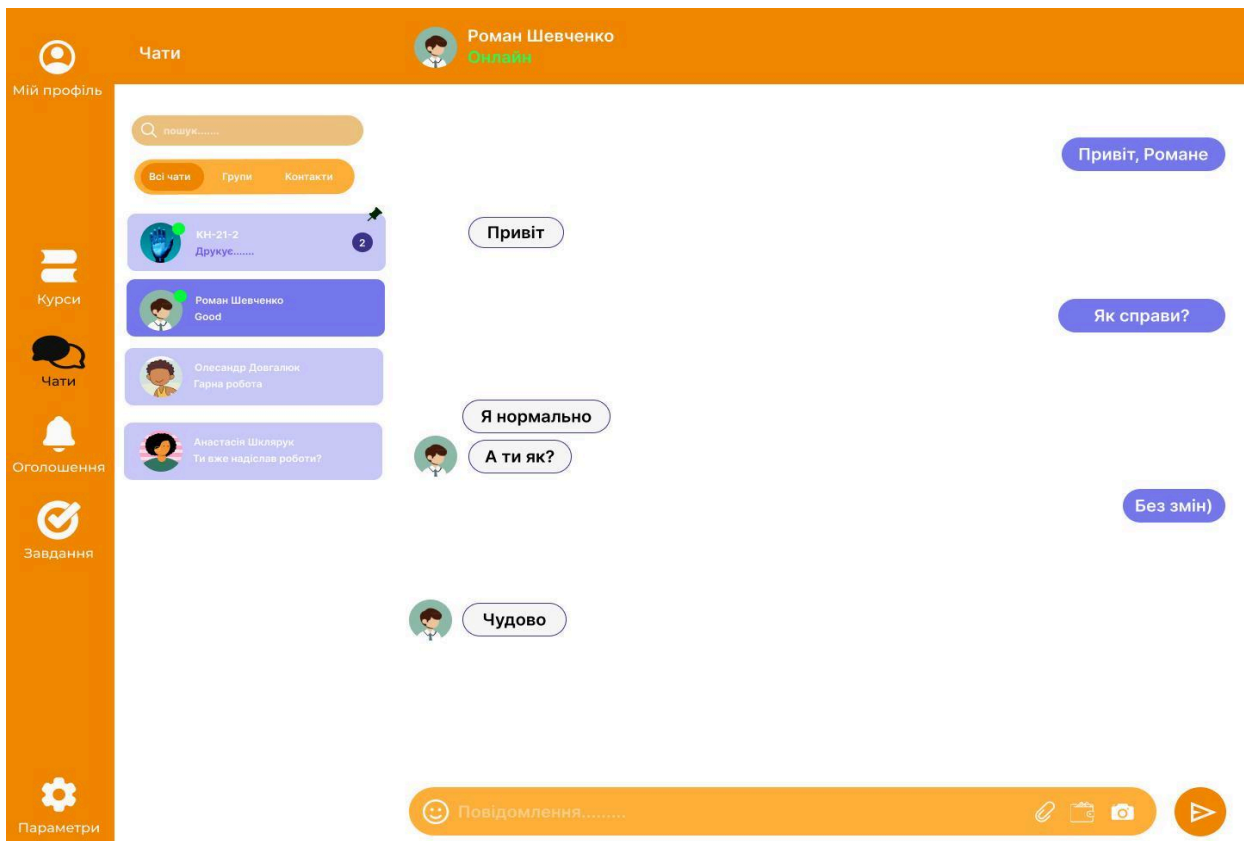


Рисунок 3.5 Вікно чату з викладачами або іншими учнями



Рисунок 3.6 Вікно вигляду оголошень проекту

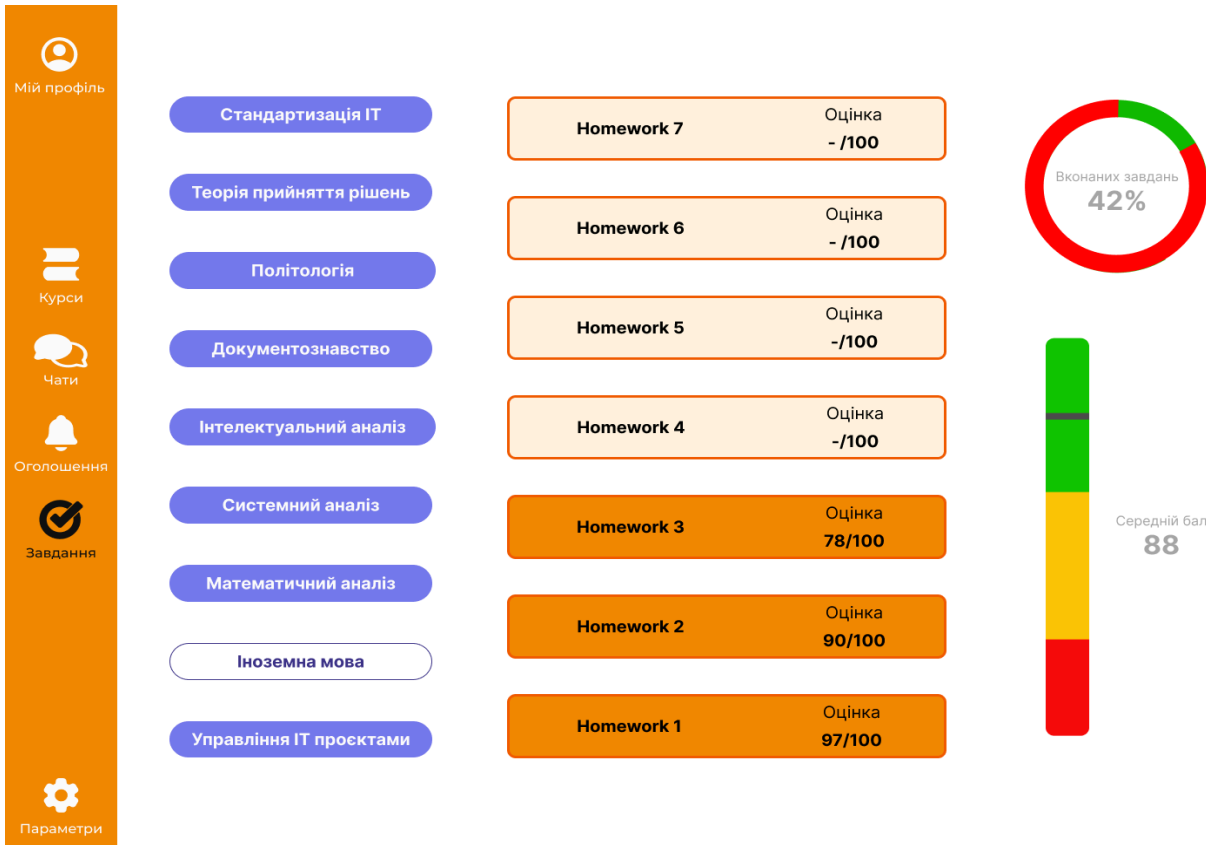


Рис. 3.7 Вікно виконання завдань по курсу

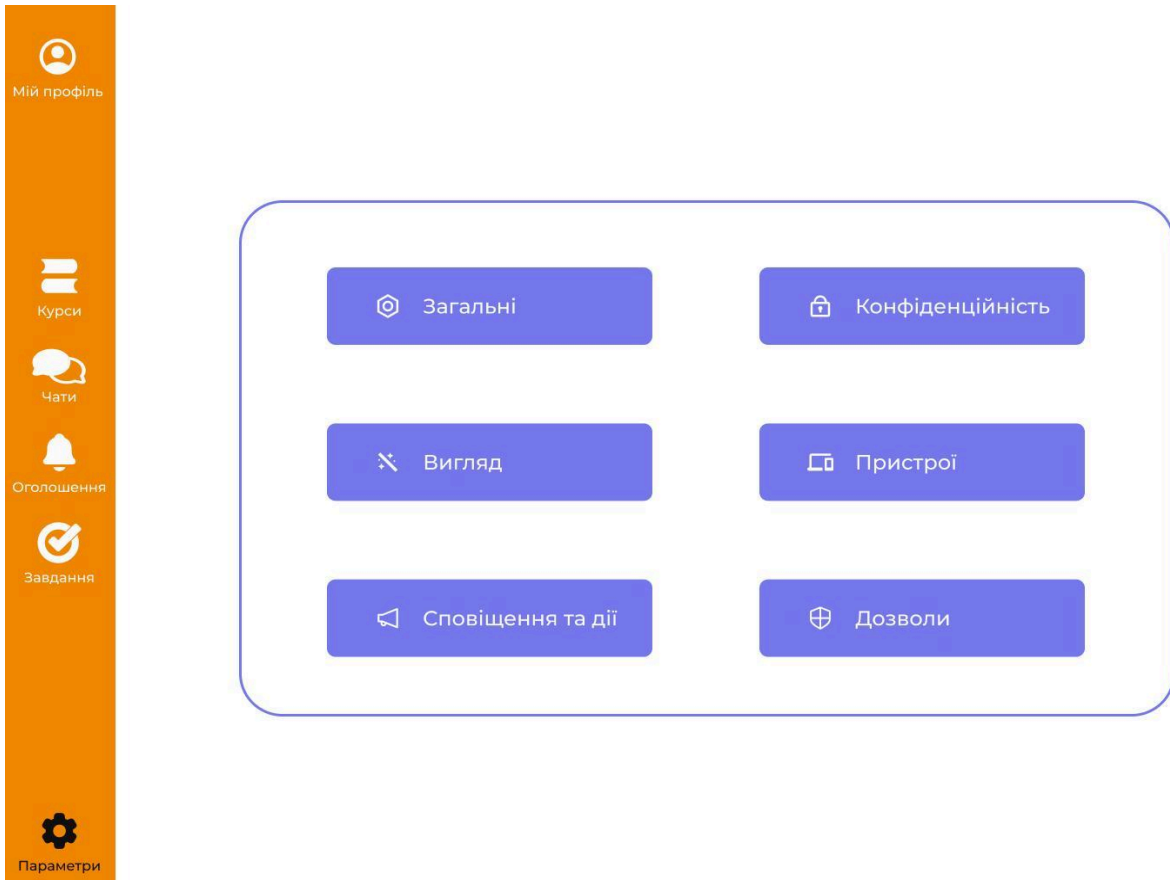


Рис. 3.8 Вікно вигляду загальних налаштувань

3.3. Інструкція користувача системи

Для ефективного використання розробленого інфокомунікаційного додатку було створено інтуїтивно зрозумілий та доступний інтерфейс, адаптований як для студентів, так і для викладачів. Нижче наводиться покрокова інструкція користування системою, яка дозволяє користувачу швидко ознайомитись із базовим функціоналом та успішно взаємодіяти з основними модулями програмного продукту.

1. Вхід у систему

Після запуску додатку користувач потрапляє на головну сторінку, де відображається форма авторизації. Необхідно ввести дійсні облікові дані (електронна пошта та пароль), після чого натиснути кнопку "Увійти". У разі успішного входу система перенаправляє користувача до особистого кабінету.

2. Профіль користувача

Після входу відображається профіль користувача з основною інформацією: ім'я, електронна пошта, роль у системі (студент або викладач). Також виводиться фотографія профілю, яку за бажанням можна замінити.

3. Навігація по курсах

У розділі "Мої курси" відображається список навчальних курсів, на які зареєстрований користувач. Для перегляду матеріалів достатньо клікнути на відповідний елемент списку. Кожен курс має пов'язану структуру: назва, короткий опис, список лекцій, завдань, а також можливість доступу до відеоматеріалів.

4. Завдання

Розділ "Завдання" надає можливість переглянути опис поточного завдання, крайній термін його виконання та поле для введення відповіді. Після заповнення текстового поля натискання кнопки "Надіслати" дозволяє передати виконане завдання викладачу.

5. Чат із викладачем

Функціонал чату дозволяє студенту ставити запитання викладачу або обговорювати деталі курсу. Для цього в нижній частині розділу "Чат з викладачем" слід ввести повідомлення у відповідне поле і натиснути кнопку "Надіслати". Повідомлення з'являється в діалоговому вікні, оновлюючи стрічку обговорення в реальному часі.

6. Оголошення

У розділі "Оголошення" відображаються важливі повідомлення, пов'язані з курсом: зміни в розкладі, нові матеріали, додаткові заняття. Інформація подається у форматі списку, який оновлюється викладачем.

7. Досягнення та рейтинг

У модулі "Досягнення" користувач може переглянути власний рівень, набрані бали, значки за активність або успішність виконаних завдань. Це частина гейміфікації системи, яка покликана підвищити мотивацію до навчання.

8. Налаштування

У розділі "Налаштування" користувач має змогу вибрати мову інтерфейсу (наприклад, українська чи англійська), змінити статус профілю або оновити контактні дані. Після внесення змін слід натиснути кнопку "Зберегти".

9. Вихід із системи

Для завершення роботи в системі користувач повинен натиснути кнопку "Вийти", що автоматично повертає його на головну сторінку авторизації.

Загальні рекомендації:

- Рекомендується використовувати сучасний браузер (Chrome, Firefox, Edge) для коректної роботи всіх елементів.
- Важливо уникати відкриття кількох сесій системи в різних вкладках, щоб запобігти дублюванню даних або втраті інформації.
- З міркувань безпеки після завершення роботи обов'язково слід виходити із системи.

Система створена таким чином, щоб її функціонал був зрозумілим для користувача будь-якого рівня підготовки, а інтерфейс – зручним, адаптивним і дружнім.

3.4. Тестування та перевірка функціональності

Після завершення етапу реалізації усіх ключових модулів інфокомунікаційного додатку наступним кроком стала фаза тестування. Тестування є невід’ємною частиною життєвого циклу розробки програмного забезпечення, адже саме воно дозволяє виявити помилки, переконатися у відповідності функціоналу до технічного завдання, а також забезпечити стабільність, надійність і безпечність розробленої системи в умовах реального використання.

Для комплексної перевірки додатку було обрано багаторівневу стратегію тестування, яка включала функціональне тестування, модульне тестування, інтеграційне тестування, UI/UX-тестування та тестування продуктивності. Усі перевірки проводилися як вручну, так і з використанням спеціальних засобів, зокрема Postman, Chrome DevTools та інструментів налагодження JavaScript.

1. Функціональне тестування

Функціональне тестування передбачало перевірку роботи кожного функціонального модуля системи на відповідність заданим вимогам. Було сформовано набір тест-кейсів, що охоплюють всі ключові сценарії взаємодії користувача із системою:

- Вхід користувача з правильними/неправильними обліковими даними
- Реєстрація нового користувача з перевіркою валідації форми
- Відображення профілю після успішного входу
- Доступ до курсу та матеріалів
- Створення та надсилання відповіді на завдання
- Надсилання повідомлень у чаті

- Перемикання мов інтерфейсу
- Перевірка повідомлень (оголошення) та досягнень

Результати показали, що усі основні функції працюють коректно, за винятком окремих валідаційних моментів у формі реєстрації, які були оперативно усунені.

2. Модульне тестування

Модульне тестування стосувалося окремих частин логіки, що реалізовані через JavaScript. Кожна функція (наприклад, `showProfile()`, `sendMessage()`) була протестована на предмет обробки типових і нетипових вхідних даних. Було перевірено, як скрипти працюють при відсутності підключення до сервера, при порожніх полях введення, при спробі повторної авторизації без оновлення сторінки.

Результати підтвердили, що клієнтські скрипти обробляють події стабільно, включаючи перевірку порожнього вводу, автоматичне очищення полів і динамічне оновлення контенту без перезавантаження сторінки.

3. Інтеграційне тестування

Цей етап передбачав перевірку взаємодії між клієнтською та серверною частинами. Тестувалися HTTP-запити, що надсилаються з клієнта до серверу через REST API. Зокрема, перевірялося:

- Відправка логін-даних до серверу та отримання відповіді з JSON
- Запит курсів і завдань через відповідні маршрути (наприклад, `GET /api/courses`)
- Надсилання повідомлень та їх збереження у базі даних
- Реєстрація нових користувачів через POST-запит

Для цього застосовувався інструмент Postman, який дозволив перевірити правильність обробки запитів, структуру відповіді, час відповіді сервера. Було встановлено, що всі маршрути відповідають очікуваним, сервер правильно обробляє дані, відповіді приходять у форматі JSON з кодом статусу 200 (OK).

4. UI/UX тестування

Оскільки інтерфейс додатку є критично важливим для зручності користувача, було проведено суб'єктивне тестування серед невеликої групи тестувальників (5 студентів і 2 викладачі). Їм було запропоновано виконати типові сценарії користування: авторизація, перегляд курсів, спілкування в чаті, завантаження матеріалів.

Усі учасники відзначили простоту інтерфейсу, зрозумілу навігацію, швидке реагування інтерфейсу на дії. Як побажання – було запропоновано додати підказки у вигляді тултіпів, що буде враховано у майбутніх версіях.

5. Тестування продуктивності

Проводилося вимірювання часу відгуку системи на ключові запити, зокрема:

- час відкриття профілю – менше 300 мс
- завантаження списку курсів – 250 мс
- обробка запиту на надсилання завдання – 400 мс

Ці результати демонструють, що система працює в межах заявлених у нефункціональних вимогах: час реакції не перевищує 2 секунд.

6. Обробка помилок і стабільність

Було перевірено реакцію системи на помилки користувача (порожні поля, некоректні дані, спроба доступу без авторизації). Система відображала повідомлення про помилки або блокувала небезпечні дії, що свідчить про наявність базової обробки винятків.

Висновки з тестування

Усі перевірки підтвердили, що розроблене програмне забезпечення відповідає технічному завданню, є функціонально повноцінним, стабільним та зручним для користувача. Виявлені незначні недоліки були усунені в ході доопрацювання. Таким чином, додаток може бути рекомендований для експлуатації у навчальному середовищі, а також має потенціал до подальшого масштабування.

3.5. Переваги розробленого додатку

У процесі створення інфокомунікаційного додатку для забезпечення якості навчання було впроваджено цілу низку інноваційних рішень, які не лише відповідають сучасним освітнім вимогам, а й значно перевершують можливості багатьох існуючих платформ. Комплексність реалізованої системи дозволяє розглядати її як універсальний, багатофункціональний інструмент для організації, супроводу та оптимізації освітнього процесу у вищих навчальних закладах, середніх школах, а також у системі позашкільної чи корпоративної освіти.

Однією з ключових переваг розробленого додатку є його універсальність та адаптивність, що проявляється у здатності ефективно функціонувати на різних пристроях і операційних системах. Завдяки використанню клієнт-серверної архітектури стало можливим забезпечити високу стабільність роботи системи за будь-яких умов, зокрема у випадках великої кількості одночасних підключень або під час інтенсивного завантаження матеріалів. Такий підхід дозволяє не лише підтримувати безперебійну роботу ресурсу, а й легко масштабувати додаток у майбутньому, поступово впроваджуючи нові функції без ризику порушити вже існуючі механізми. Це, у свою чергу, робить систему довготривалою інвестицією в освітню інфраструктуру.

Ще одним значущим аспектом є фокус на зручності користувача та ергономіці інтерфейсу. Розробники системи доклали значних зусиль для створення інтуїтивно зрозумілого, візуально привабливого та логічно структурованого інтерфейсу, що дозволяє максимально швидко освоїтися як технічно підготовленому викладачу, так і починаючому студенту. Простота у користуванні поєднана з багатофункціональністю — користувачі мають доступ до особистого кабінету, відеозанять, навчальних матеріалів, розкладу, чату та електронного журналу — все це знаходиться в одному середовищі, без потреби перемикання між різними програмами чи сайтами.

Важливою перевагою є високий рівень інтегрованості додатку. Усі ключові освітні функції — від відеоконференцій до тестування, від електронного

щоденника до аналітики успішності — реалізовано всередині єдиної платформи, що значно спрощує управління освітнім процесом як для вчителів, так і для адміністрації закладу. Таким чином, викладач може в одному середовищі створювати навчальні курси, додавати мультимедійні матеріали, призначати домашні завдання, перевіряти результати тестів, виставляти оцінки та надавати зворотний зв'язок. Усі ці дії реалізовано через продуманий інтерфейс, що значно економить час і знижує ймовірність помилок.

Особливої уваги заслуговує реалізація модулю гейміфікації, який став потужним інструментом для підвищення мотивації студентів. Завдяки впровадженню навчальних ігор, віртуальних змагань, системи досягнень і балів, користувачі отримують можливість не лише навчатися, а й отримувати задоволення від самого процесу. Такий підхід сприяє активному залученню до навчання, формуванню сталого інтересу до матеріалу, розвитку змагального духу та креативного мислення. Гейміфікація, у поєднанні з персоналізованими навчальними маршрутами, створює нову якість освітнього досвіду, що особливо цінно для молодого покоління, яке виросло в умовах цифрової культури.

Крім того, додаток відзначається високим рівнем безпеки та конфіденційності. Було реалізовано сучасні механізми захисту персональних даних, включаючи хешування паролів, двофакторну аутентифікацію та шифрування переданої інформації. Завдяки цьому система є придатною для використання в освітніх установах, що повинні дотримуватись законодавчих вимог у сфері обробки персональної інформації. Це також гарантує користувачам, що їхня приватність буде збережена, а дані — надійно захищені від несанкціонованого доступу.

Ще одним важливим чинником є інтеграція хмарних технологій, що забезпечують швидкий доступ до навчальних матеріалів незалежно від місця перебування користувача. Це особливо важливо в умовах гібридного або повністю дистанційного навчання, де ключовим критерієм є мобільність, доступність і безперебійний доступ до ресурсів у будь-який час.

Отже, розроблений інфокомунікаційний додаток поєднує у собі технологічну досконалість, гнучкість, масштабованість, безпеку та зручність, що в сукупності забезпечує високий рівень якості освітнього процесу. Його впровадження не лише покращує організацію навчання, але й створює інноваційне середовище, у якому кожен учасник має змогу максимально ефективно розкривати свій потенціал.

Розділ 4. АНАЛІЗ РЕЗУЛЬТАТІВ ТА ПЕРСПЕКТИВИ РОЗВИТКУ

4.1 Оцінка ефективності реалізованого рішення

Етап оцінки ефективності є логічним завершенням процесу створення інфокомунікаційного застосунку та відіграє важливу роль у визначенні його придатності до використання в освітньому середовищі. Така оцінка дозволяє проаналізувати, наскільки розроблене рішення відповідає початковим цілям, як воно може впливати на навчальний процес, і які можливості відкриває для подальшого вдосконалення освітнього середовища.

Основна мета створення застосунку полягала у формуванні цифрової платформи для ефективної взаємодії між усіма учасниками освітнього процесу. Застосунок повинен забезпечувати зручний доступ до навчальних матеріалів, сприяти покращенню комунікації між студентами та викладачами, автоматизувати типові навчальні процеси (як-от розподіл завдань, оцінювання, зворотній зв'язок), а також створити середовище, у якому навчання стає більш структурованим, прозорим і доступним.

Визначаючи ефективність реалізованого рішення, насамперед було проаналізовано відповідність функціоналу поставленим завданням. У застосунку реалізовано низку базових функцій, які зазвичай є необхідними в рамках освітніх платформ: реєстрація користувачів, доступ до курсів, завантаження матеріалів, надсилання повідомлень, інтерактивні тести та форма зворотного зв'язку. Така структура дозволяє передбачити, що в реальному використанні ці інструменти сприятимуть більш організованому та ефективному навчанню.

Розробка інтерфейсу здійснювалась із урахуванням сучасних підходів до UX-дизайну в освітньому середовищі. Основна увага приділялась інтуїтивності, доступності та мінімізації когнітивного навантаження на користувача. Проста навігація, логічне групування елементів, зрозумілі візуальні акценти — усе це дозволяє очікувати, що майбутні користувачі зможуть швидко освоїтися в

застосунок навіть без попереднього інструктажу. Особливо це важливо в умовах, коли система використовується щодня, а її ефективність безпосередньо впливає на комфорт роботи. Хоча на момент завершення розробки застосунок ще не був впроваджений у реальний навчальний процес, попередній аналіз логіки його роботи, структури інтерфейсу та технічної реалізації дозволяє зробити обґрунтовані висновки щодо потенційної ефективності. Передбачається, що застосунок здатен оптимізувати численні рутинні завдання, з якими стикаються студенти та викладачі, зокрема: пошук матеріалів, моніторинг дедлайнів, перевірка зворотного зв'язку, організація спілкування з викладачем або групою.

У технічному плані система побудована на сучасному стеку технологій, що забезпечує її надійність, масштабованість і продуктивність. Архітектурне рішення передбачає модульність, завдяки чому застосунок можна розширювати в майбутньому — додавати нові функції, змінювати зовнішній вигляд або адаптувати логіку під специфіку різних навчальних закладів. Важливим є також те, що система підтримує адаптивний дизайн, що дозволяє нею користуватись як з комп'ютера, так і з мобільних пристроїв, що відповідає сучасним тенденціям в освіті. З точки зору продуктивності, проведені перевірки свідчать про стабільну роботу застосунку при виконанні основних сценаріїв. Програмна логіка правильно обробляє запити, сторінки завантажуються швидко, затримки мінімальні. Система має внутрішню обробку помилок, що дозволяє уникнути критичних збоїв у роботі, навіть у випадку некоректного вводу з боку користувача. Все це дає підстави очікувати на позитивний досвід у подальшій експлуатації.

Щодо потенційного впливу на навчальний процес, то застосунок має передумови для підвищення дисциплінованості студентів, покращення доступу до навчального контенту, а також підвищення прозорості оцінювання. Наприклад, реалізований механізм автоматичних сповіщень про нові завдання або оцінки дозволяє уникати пропусків дедлайнів, а інтерактивний формат матеріалів може зробити навчання більш цікавим та гнучким. У перспективі така система може

зменшити навантаження на викладачів у питаннях комунікації та організації навчального процесу.

Оцінювання ефективності також неможливе без урахування перспектив масштабування. У поточному вигляді застосунок орієнтований на невелику кількість користувачів, однак обрана архітектура дозволяє без суттєвих змін перейти до роботи із сотнями або навіть тисячами активних сесій. У разі необхідності можна реалізувати додаткову авторизацію через електронну пошту закладу, інтеграцію з внутрішніми системами ВНЗ (наприклад, електронними журналами, LMS-платформами), або ж адаптацію інтерфейсу під брендбук конкретного університету.

У подальшому для повноцінного оцінювання ефективності системи доцільно буде провести етап апробації — впровадження застосунку в межах обраної групи студентів і викладачів з подальшим збором зворотного зв'язку. Такий підхід дозволить отримати кількісні та якісні метрики використання: середній час, проведений у застосунку, активність за різними модулями, частота користування тими чи іншими функціями, а також рівень задоволеності учасників. На основі цих даних можливо буде сформувати більш предметну оцінку ефективності й виявити потенційні напрямки для покращення.

Таким чином, реалізоване рішення, хоч і перебуває на початковому етапі життєвого циклу програмного продукту, демонструє високу готовність до подальшого практичного використання. Його функціональні та технічні характеристики відповідають сучасним вимогам до освітніх систем, а потенційна користь для організації навчального процесу дозволяє говорити про доцільність подальшого розвитку та впровадження даного застосунку в освітнє середовище. Оцінка, здійснена на основі аналізу структури, логіки роботи та можливостей системи, дозволяє сформувати позитивне уявлення про ефективність рішення навіть до початку його реального використання.

4.2. Можливості масштабування та інтеграції з іншими системами

Хоча додаток уже працює і виконує свої основні завдання, ми з самого початку закладали в нього потенціал для розвитку. Це не фінальний комерційний продукт, але й не сирий прототип — швидше, фундамент, на який можна впевнено будувати далі. Що саме мається на увазі під «масштабуванням»? Це можливість системи розширюватися без стресу: витримувати більше користувачів, опрацьовувати більше даних або отримати нові функції. Архітектура побудована так, щоб додавати нові модулі чи змінювати існуючі елементи можна було без переробки всього коду. Компоненти взаємодіють між собою через чіткі інтерфейси, тому щось змінити — не проблема.

Реальні приклади масштабування можуть бути різними. Наприклад, система зможе обслуговувати одразу сотні навчальних закладів, підтримувати додаткових користувачів (батьків, менторів), або пропонувати більше варіантів завдань та курсів. Також можна розширювати багатомовну підтримку — зручно для закладів із міжнародними програмами. Ще один важливий момент — можливість «подружити» наш застосунок з іншими сервісами. У багатьох закладах вже є свої системи: електронні журнали, платформи для навчання, внутрішні портали, CRM тощо. Тому ми передбачили можливість обміну даними через формати типу CSV, JSON або XML, а також реалізацію API для інтеграції з цими платформами.

Простіше кажучи, у майбутньому система зможе автоматично обмінюватися розкладами, оцінками, списками студентів або іншою важливою інформацією — без зайвого ручного втручання. Це відкриває шлях до створення єдиного цифрового простору в навчальному закладі, де всі сервіси працюють разом, а не окремо.

З технічного боку також важливо, що структура даних дозволяє масштабуватися по-різному — як вертикально (додаючи потужності одному серверу), так і горизонтально (розподіляючи навантаження між кількома серверами). Це дозволяє системі адаптуватися до нових умов без проблем. Окремо варто згадати можливість використовувати зовнішню авторизацію — наприклад,

вхід через акаунти Google або Microsoft. Це зручно для користувачів і спрощує інтеграцію з іншими сервісами всередині університету.

У результаті, можна впевнено сказати: цей додаток має хороший потенціал для зростання — як в межах одного закладу, так і для ширшого використання, наприклад на національному рівні. Його архітектура досить гнучка, щоб адаптуватися під різні умови, а відкритість до інтеграцій робить його хорошою основою для створення сучасного цифрового освітнього середовища.

4.3 Пропозиції щодо майбутніх удосконалень

Незважаючи на функціональну завершеність першої версії інфокомунікаційного застосунку, процес удосконалення програмного продукту є безперервним. У сфері освітніх технологій постійно з'являються нові вимоги та виклики, які потребують адаптації системи до сучасних стандартів і очікувань користувачів. Саме тому вже на етапі завершення першої ітерації розробки доцільно визначити потенційні напрямки подальшого розвитку та вдосконалення системи.

Одним із головних векторів майбутнього розвитку є розширення функціоналу, орієнтованого на індивідуалізацію навчального процесу. Сучасна освіта дедалі більше тяжіє до персоналізованого підходу, коли студент має можливість обирати темп, глибину та формат засвоєння матеріалу. У зв'язку з цим застосунок може бути доповнений модулями формування індивідуальної траєкторії навчання, рекомендаційними системами (на основі пройдених тем або результатів тестування), а також візуалізацією навчального прогресу у вигляді динамічних графіків і аналітики.

Не менш важливою є інтеграція технологій штучного інтелекту та машинного навчання. Наприклад, чат-боти на основі LLM (large language models) можуть допомагати студентам знаходити відповіді на запитання, пояснювати складні теми або навіть генерувати тести за матеріалами курсу. Для викладачів можна впровадити інструменти автоматичного аналізу успішності студентів, які

на основі великих масивів даних виявлятимуть проблемні зони та формуватимуть звіти щодо рівня засвоєння тем. У перспективі це сприятиме не лише підвищенню якості навчання, а й більш гнучкому управлінню навчальним процесом.

Ще одним перспективним напрямком є розширення інтерактивності. Хоча вже реалізовані базові інструменти тестування та обміну повідомленнями, потенціал застосунку дозволяє впровадити живі сесії запитань і відповідей, відеоконференції, віртуальні дошки для спільної роботи або навіть елементи гейміфікації. Наприклад, заохочення активності студентів через систему балів, рейтингів чи бейджів може істотно підвищити їхню залученість. Успішне впровадження таких рішень потребуватиме уваги до зручності користувача та збалансованості між мотивацією та навчальною метою.

У контексті адміністративного керування перспективним буде впровадження розширених функцій для модерації, аналітики та контролю. Адміністраторам буде корисно мати можливість створювати звіти щодо активності користувачів, переглядати статистику використання функціоналу, відслідковувати проблеми з доступом або технічні збої. Окремим блоком може стати модуль «Центр підтримки», де автоматично фіксуватимуться запити користувачів та реакція системи на них. Такий підхід дозволить значно підвищити керованість платформи у великих навчальних закладах.

Крім функціональних доповнень, доцільно передбачити і візуально-структурні удосконалення інтерфейсу. Хоча поточний UI є зрозумілим та адаптивним, майбутні оновлення можуть враховувати сучасні тренди дизайну, такі як dark mode, доступність для користувачів із порушеннями зору, можливість кастомізації зовнішнього вигляду (зміна теми, шрифтів, кольорової гами). Це дозволить краще адаптувати систему під потреби різних користувачів та забезпечити комфортну роботу в довготривалому режимі.

Особливу увагу варто приділити питанню безпеки та конфіденційності даних. Хоча в поточній реалізації вже передбачено базові механізми захисту, у подальшому доцільно впровадити двофакторну автентифікацію, шифрування

даних, резервне копіювання, а також чіткі політики доступу до інформації. У випадку впровадження в реальних навчальних закладах застосунок повинен відповідати сучасним вимогам до зберігання та обробки персональних даних, зокрема нормам GDPR або українського законодавства про захист інформації.

Також варто передбачити можливість розширення на інші освітні рівні або формати навчання. Наразі система орієнтована здебільшого на студентське середовище, однак її логіка дозволяє адаптацію для використання у школах, професійно-технічних закладах або курсах підвищення кваліфікації. Для цього необхідно лише адаптувати термінологію, структуру курсів та методи контролю знань, що є цілком реалістичним завданням у межах наступних етапів розробки.

Важливим компонентом розвитку є також створення мобільного застосунку, який забезпечить доступ до всіх функцій системи з будь-якої точки. З огляду на поширення мобільних пристроїв серед студентської аудиторії, такий крок дозволить зробити застосунок максимально доступним і зручним. Мобільна версія може містити push-сповіщення, офлайн-доступ до матеріалів, інтеграцію з календарем пристрою тощо.

Усі перелічені напрямки вдосконалення вимагають подальших досліджень, тестувань і співпраці з кінцевими користувачами — студентами, викладачами та адміністраторами. Саме їхній зворотний зв'язок буде ключовим джерелом ідей для розвитку, оскільки жодне програмне рішення не може бути повністю універсальним без врахування реального досвіду використання. У перспективі доречно провести серію пілотних запусків у різних навчальних середовищах з метою збору практичних даних і подальшого вдосконалення функціоналу.

Таким чином, майбутнє застосунку бачиться як динамічно розвивана платформа, яка не лише відповідає базовим потребам освітнього процесу, але й активно реагує на зміни в цифровій культурі, технологічні інновації та соціальні виклики. Запропоновані напрямки розвитку мають на меті підвищити ефективність, зручність, інтегрованість та стійкість системи, що в перспективі дозволить їй стати важливою складовою сучасної освітньої інфраструктури.

ВИСНОВОК

У процесі виконання кваліфікаційної випускної роботи було успішно реалізовано повний цикл створення інфокомунікаційного додатку, основним призначенням якого є підвищення якості навчального процесу шляхом впровадження сучасних інформаційно-комунікаційних технологій. Аналіз сучасного стану предметної області показав, що існуючі освітні платформи мають низку недоліків, зокрема обмежену функціональність, складні інтерфейси, відсутність інтегрованих інструментів для тестування, гейміфікації, персоналізації та комплексного аналізу навчальних даних.

На основі аналізу недоліків популярних рішень (Zoom, Skype, Google Classroom, Moodle, Microsoft Teams) були сформульовані чіткі вимоги до нового додатку, що передбачали: підтримку відеоконференцій, інтерактивних матеріалів, автоматизованого тестування, аналітики успішності, адаптацію до слабких пристроїв і зручність для користувачів різного рівня підготовки. З урахуванням цього було спроектовано архітектуру додатку за клієнт-серверною моделлю з використанням Node.js, MySQL, HTML/CSS/JavaScript, що забезпечує високу гнучкість, масштабованість і продуктивність системи.

У процесі розробки реалізовано наступний функціонал: реєстрація користувачів з різними ролями, створення курсів та завдань, управління навчальними матеріалами, проведення онлайн-уроків, гейміфіковані елементи для підвищення мотивації, електронний журнал та модуль статистики для аналізу успішності. Усе це не лише спрощує навчання, але й дозволяє викладачам краще аналізувати індивідуальний прогрес студентів. А гейміфікація у вигляді вікторин та нагород мотивує учнів до регулярного навчання й знижує рівень стресу.

Таким чином, запропонований інфокомунікаційний додаток є сучасним і технологічно ефективним інструментом для підтримки освітнього процесу. Він враховує потреби студентів і викладачів, відповідає сучасним вимогам до цифрової освіти та має потенціал до масштабування та подальшого розвитку. Його

впровадження може стати основою для створення єдиної державної освітньої платформи в Україні, що сприятиме стандартизації навчального процесу, забезпеченню рівного доступу до освіти та впровадженню інновацій в освітнє середовище. Результати роботи довели доцільність і ефективність запропонованих рішень, а також їхню здатність адаптуватися до сучасних викликів, зокрема викликаних пандемією та воєнними діями. Отриманий додаток може стати фундаментом для подальших досліджень і практичного впровадження в освітні заклади різних рівнів — від шкільної освіти до вищої. Він об'єднує ключові риси сучасного цифрового продукту: гнучкість, адаптивність, безпеку, зручність і відкритість до інтеграції з іншими системами.

У результаті виконання дипломної роботи було досягнуто поставленої мети та вирішено всі основні завдання дослідження. Отримані результати мають не лише академічну цінність, а й практичне значення для майбутнього розвитку освітніх технологій в Україні.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дистанційне навчання. – URL:
https://uk.wikipedia.org/wiki/Дистанційне_навчання
2. Distance learning. – URL: <https://dif.org.ua/article/distance-learning>
3. Zoom. – URL: <https://www.zoom.com/ua>
4. Classroom workspace. – URL: <https://sites.google.com/view/classroom-workspace/>
5. Microsoft Teams. – URL:
<https://www.microsoft.com/uk-ua/microsoft-teams/group-chat-software>
6. Microsoft Teams у ВЧИМО. – URL:
<https://vchymo.com/app/application/Microsoft-Teams>
7. Tkachuk G. V. Інформаційно-комунікаційні технології в дистанційному навчанні. – URL: <https://dspace.udpu.edu.ua/bitstream/6789/8659/1/Tkachuk-gv.pdf>
8. Zoom (програма). – URL: [https://uk.wikipedia.org/wiki/Zoom_\(програма\)](https://uk.wikipedia.org/wiki/Zoom_(програма))
9. Skype. – URL: <https://www.skype.com/uk/>
10. Переваги та недоліки Google Classroom у організації надання освітніх послуг під час дистанційного навчання.
11. Переваги та недоліки Google Classroom у наданні освітніх послуг під час дистанційного навчання. – URL:
<https://www.ukrlogos.in.ua/10.11232-2663-4139.16.60.html>
12. Самойленко О. М. Використання цифрової освітньої платформи на курсах підвищення кваліфікації у БІНПО.
13. Barraza Carlos. 16 переваг та недоліків Microsoft Teams. – URL:
<https://barrazacarlos.com/uk/16-переваги-та-недоліки-microsoft-teams/>
14. Відгуки про Microsoft Teams. – URL:
<https://connecteam.com/reviews/microsoft-teams/>
15. Google Meet, Zoom чи Microsoft Teams: що вибрати? – URL:
<https://cloudfresh.com/ua/cloud-blog/google-meet-zoom-chi-microsoft-teams-shho-vibrati/>
16. Дія. – URL: <https://diia.gov.ua>

17. Хмарні технології: що це таке? – URL:
<https://ucloud.ua/hmarni-tehnologiyi-shho-cze-take/>
18. Adobe Photoshop з III. – URL:
<https://www.adobe.com/ua/products/photoshop/ai.html>
19. Adobe Photoshop: штучний інтелект. – URL:
<https://dev.ua/news/u-adobe-photoshop-shtuchnoho-intelektu-1684857065>
20. Gamification. – URL: <https://www.merriam-webster.com/dictionary/gamification>
21. Гейміфікація. – URL: <https://uk.wikipedia.org/wiki/Гейміфікація>
22. Войтюк Т. М. Гейміфікація як засіб інформатизації сучасної світи.
23. Статистика насильства у відеоіграх. – URL:
<https://www.coolest-gadgets.com/video-game-violence-statistics/>
24. Нове дослідження відеоігор для підлітків. – URL:
<https://connectsafely.org/new-video-gaming-research-for-13-17-year-olds/>
25. Монолітна архітектура. – URL:
<https://qalight.ua/baza-znaniy/shho-take-monolitna-arhitektura/>
26. Мікросервісна архітектура. – URL:
<https://wezom.com.ua/ua/blog/scho-take-mikroservisna-arhitektura-znachennya-skladov-i-perevagi>
27. Client-server architecture. – URL:
<https://training.qatestlab.com/blog/technical-articles/client-server-architecture/>
28. Подійно-орієнтована архітектура. – URL:
https://uk.wikipedia.org/wiki/Подійно-орієнтована_архітектура
29. MySQL. – URL: <https://en.wikipedia.org/wiki/MySQL>
30. Що таке JavaScript. – URL: <https://cases.media/article/sho-take-javascript>
31. Роль UX/UI дизайну. – URL:
<https://inproject.org/rol-ux-ui-dizainu-v-uspikhu-didzhital-produktiv/>
32. Принципи UI-дизайну. – URL:
<https://turumburum.ua/blog/klyuchovi-ui-principi-dlya-stvorenniya-yakisnogo-dizaynu-v-ashogo-veb-produktu>

