

**Міністерство освіти і науки України**  
**Луцький національний технічний університет**  
**Факультет комп'ютерних та інформаційних технологій**  
**Кафедра комп'ютерних наук**

**КВАЛІФІКАЦІЙНА РОБОТА**  
**ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «МАГІСТР»**

**РОЗРОБКА ТА ДОСЛІДЖЕННЯ ВЕБ-ПЛАТФОРМИ**  
**ДЛЯ ЕФЕКТИВНОГО НАДАННЯ ОНЛАЙН-КУРСІВ**  
**З ІСПАНСЬКОЇ МОВИ**

**RESEARCH AND DEVELOPMENT OF A WEB PLATFORM FOR**  
**EFFECTIVE PROVISION OF ONLINE SPANISH LANGUAGE COURSES**

спеціальність 122 Комп'ютерні науки

освітня програма «Комп'ютерні науки»

Виконала: здобувачка вищої освіти  
групи КНм-21  
Васюхник Софія Олександрівна

\_\_\_\_\_

(підпис)

Керівник: к.т.н., доцент  
Лук'янчук Юрій Анатолійович

\_\_\_\_\_

(підпис)

Кваліфікаційну роботу  
допущено до захисту  
«\_\_\_» \_\_\_\_\_ 2025 р.  
Гарант освітньої програми:  
к.т.н., доцент  
Ліщина Валерій Олександрович

\_\_\_\_\_

(підпис)

Луцьк – 2025 року

**ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**

Факультет *комп'ютерних та інформаційних технологій*

Кафедра *комп'ютерних наук*

Ступінь вищої освіти: *магістр*

Галузь знань: *12 Інформаційні технології*

Спеціальність: *122 Комп'ютерні науки*

Освітня програма: *«Комп'ютерні науки»*

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Валерій ЛІЩИНА

«14» травня 2025 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА  
ДРУГОГО (МАГІСТЕРСЬКОГО) РІВНЯ ВИЩОЇ ОСВІТИ**

Васюхник Софія Олександрівна

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи «Розробка та дослідження веб-платформи для ефективного надання онлайн-курсів з іспанської мови»

Керівник к.т.н., доцент Лук'янчук Юрій Анатолійович

затверджені наказом закладу вищої освіти від «14» *травня 2025* р. № 255/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи «05» *грудня 2025* р.

3. Вихідні дані до роботи: \_\_\_\_\_

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити):

Аналіз сучасного стану проблеми, існуючих методів і засобів її розв'язання, аналіз і вибір засобів проектування, опис функціонального наповнення об'єкта проектування, розробка й обґрунтування системного наповнення, експериментальне дослідження результативності предмету дослідження.

5. Перелік графічного матеріалу: 18 рисунків, 6 таблиць та 3 додатки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Аналіз проблематики за темою роботи та постановка завдань дослідження</i>	<i>Лук'янчук Ю. А.</i>		
<i>Теоретичне дослідження та практична реалізація предмету дослідження</i>	<i>Лук'янчук Ю. А.</i>		
<i>Експериментальне дослідження результативності предмету дослідження</i>	<i>Лук'янчук Ю. А.</i>		
<i>Показник запозичень тексту</i>		_____ %	
<i>Інструментальна перевірка</i>	<i>Кошелюк В. А.</i>		
<i>Нормоконтроль</i>	<i>Сачук В. О.</i>		
<i>Гарант ОПП</i>	<i>Лищина В. О.</i>		

7. Дата видачі завдання «14» травня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	<i>Провести огляд літературних джерел по темі кваліфікаційної роботи</i>	<i>до 30.06.2025 р</i>	
2	<i>Провести аналіз загальної проблеми і вибір напрямків дослідження</i>	<i>до 01.09.2025 р.</i>	
3	<i>Розробити функціональну схему роботи програмного продукту</i>	<i>до 01.10.2025 р</i>	
4	<i>Описати засоби розробки об'єкта проектування</i>	<i>до 15.10.2025 р.</i>	
5	<i>Практична реалізація об'єкта проектування</i>	<i>до 10.11.2025 р.</i>	
6	<i>Провести експериментальне дослідження результативності предмету дослідження</i>	<i>до 25.11.2025 р.</i>	
7	<i>Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедрі</i>	<i>до 05.12.2025 р.</i>	

Здобувач вищої освіти \_\_\_\_\_ Софія ВАСЮХНИК

Керівник роботи \_\_\_\_\_ Юрій ЛУК'ЯНЧУК

## АНОТАЦІЯ

Васюхник С. О. Дослідження та розробка веб-платформи для ефективного надання онлайн-курсів з іспанської мови. Рукопис.

Кваліфікаційна робота магістра ОП «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки». Луцький національний технічний університет. Луцьк, 2025.

Кваліфікаційна робота магістра складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. Загальний обсяг роботи становить 91 сторінок, містить 18 рисунків, 6 таблиць та 3 додатки.

Кваліфікаційна робота магістра присвячена дослідженню сучасних підходів до організації онлайн-навчання іноземних мов та розробці веб-платформи для надання курсів іспанської мови. Проаналізовано стан онлайн-освіти в Україні та світі, виявлено тенденції зростання глобального ринку та збільшення попиту на вивчення іноземних мов. Досліджено особливості викладання іспанської мови в дистанційному форматі, обґрунтовано вибір методу та інтеграції інтерактивних інструментів для забезпечення ефективної взаємодії. Проведено аналіз існуючих засобів і методів і засобів веб-проекування. Виконано порівняльний аналіз існуючих мовних платформ, визначено їх переваги та недоліки, сформовано функціональні й нефункціональні вимоги до системи. На основі методології Rational Unified Process здійснено UML-моделювання з побудовою діаграм варіантів використання, активності та класів.

Ключові слова: веб-платформа, онлайн-курси, дистанційне навчання, UML-моделювання, RUP, веб-розробка, іспанська мова, React, Node.js, MongoDB.

## ABSTRACT

Sofiia Vasiukhnyk. Research and development of a web platform for effective delivery of online Spanish language courses. Manuscript.

Master's Qualification Thesis, Educational Program «Computer Science», specialty 122 «Computer Science». Lutsk National Technical University. Lutsk, 2025.

The master's thesis consists of an introduction, three chapters, conclusions, a list of references, and appendices. The total volume of the thesis is 91 pages and includes 18 figures, 6 tables, and 3 appendices.

The master's thesis is devoted to the study of modern approaches to organizing online foreign language learning and the development of a web platform for delivering Spanish language courses. The state of online education in Ukraine and worldwide is analyzed, revealing trends in the growth of the global market and the increasing demand for foreign language learning. The specific features of teaching Spanish in a distance format are examined, and the choice of the method as well as the integration of interactive tools to ensure effective learner-teacher interaction is justified. An analysis of existing tools and methods of web development is conducted. A comparative analysis of existing language-learning platforms is performed, their advantages and disadvantages are identified, and functional and non-functional requirements for the system are defined. Based on the Rational Unified Process methodology, UML modeling was carried out, including use case diagrams, activity diagrams, and class diagrams.

Keywords: web platform, online courses, e-learning, UML modeling, RUP, web development, Spanish language, React, Node.js, MongoDB.

## ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ, МЕТОДІВ РОЗРОБКИ ТА ПОСТАНОВКА ЗАВДАННЯ .....	11
1.1 Стан онлайн-освіти в Україні та світі .....	11
1.2 Особливості викладання іспанської мови онлайн .....	13
1.3 Аналіз існуючих мовних платформ для надання курсів з іспанської та формування вимог до веб-платформи.....	18
1.4 Огляд і аналіз методів та засобів розробки веб-платформ .....	22
1.5 Постановка завдання на кваліфікаційну роботу магістра.....	25
РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВЕБ-ПЛАТФОРМИ З НАДАННЯ КУРСІВ ІСПАНСЬКОЇ МОВИ.....	27
2.1 Вибір методології та інструментів проєктування.....	27
2.2 UML-моделювання системи.....	32
2.3 Обґрунтування вибору шляхів, технологій та засобів вирішення поставленого завдання.....	40
2.4 Практична реалізація веб платформи для надання курсів з іспанської.....	43
2.5 Оцінка функціональних характеристик системи .....	49
РОЗДІЛ 3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ РЕЗУЛЬТАТИВНОСТІ РОЗРОБЛЕНОЇ ВЕБ-ПЛАТФОРМИ.....	52
3.1 Методика проведення дослідження .....	52
3.2 Обробка та аналіз результатів дослідження .....	54
ВИСНОВКИ.....	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	67
ДОДАТКИ.....	70

## ВСТУП

У сучасному світі цифрові технології дедалі активніше інтегруються у всі сфери життя, зокрема й в освіту. Онлайн-навчання стало не лише трендом, а й невід'ємною складовою для тисяч учнів та викладачів, які щодня використовують цифрові інструменти для передачі знань. Особливої важливості ця тема набуває в контексті вивчення іноземних мов, де ефективна комунікація, інтерактивність та постійна взаємодія з учнями мають вирішальне значення.

Вивчення іспанської мови як однієї з найпоширеніших мов світу потребує гнучкого та зручного середовища, яке могло б поєднати в собі функції представлення структури, презентації курсів, інтеграцію навчання, визначення свого рівня іноземної, і легкого запису на заняття та налагодження зворотного зв'язку для зручної взаємодії з потенційними та вже наявними учнями.

Актуальність цієї теми зумовлена зростанням кількості онлайн мовних платформ та збільшенням попиту на вивчення іноземної мови серед українців різних категорій населення. Загалом, український ринок завжди мав попит на вивчення іноземних мов, проте в зв'язку з подіями останніх років, зокрема з початком повномасштабного вторгнення попит на вивчення суттєво виріс на фоні значної міграції населення.

Окрім освітніх та демографічних факторів, варто відзначити й економічні тенденції. Аналізуючи сучасний комерційний ринок можна зазначити, що як в Україні, як і в усьому світі, спостерігається активна цифровізація бізнесу з метою залучення більшої кількості клієнтів, ефективно представлення послуг та самопрезентації.

Все більше компаній намагаються перенести свої послуги в онлайн-простір, адже саме цифрова присутність часто визначає конкурентоспроможність і сучасність бізнесу – це вже не просто модний тренд, а необхідність. Бізнес, який не має цифрового представлення, сьогодні сприймається як такий, що відстає від часу.

На сьогодні існує велика кількість мовних онлайн-платформ, однак більшість із них орієнтовані на масову аудиторію і не враховують потреб окремих навчальних центрів. Значна частина таких систем має обмежену інтерактивність або складний інтерфейс. Тому актуальною задачею є створення веб-платформи, адаптованої під конкретні освітні цілі та потреби користувачів, із застосуванням сучасних фреймворків та технологій розробки.

Метою магістерської роботи є дослідження, моделювання та програмна реалізація веб-платформи для надання курсів іспанської мови, яка дозволить централізувати управління навчальним процесом та підвищити ефективність його організації.

Об'єктом дослідження є процес організації онлайн-курсів з вивчення іспанської мови.

Предметом дослідження є програмні засоби, принципи та інструменти розробки веб-платформ для онлайн-навчання.

У межах роботи передбачається виконати такі основні завдання:

- проаналізувати наявні аспекти онлайн-навчання, а також наявні технічні рішення в контексті мовних курсів;
- визначити та формалізувати вимоги до веб-платформи, включаючи структуру, основні функціональні можливості та критерії якості;
- обґрунтувати вибір технологічного стеку та архітектурних рішень, придатних для реалізації платформи мовної школи;
- розробити UML-моделі, що відображають логіку роботи системи, взаємодію користувачів та структуру даних;
- реалізувати веб-платформу відповідно до визначених вимог, включаючи модулі презентації курсів, визначення рівня, запису на заняття та зворотного зв'язку;
- інтегрувати можливість проходження курсів;
- провести тестування функціональних компонентів та оцінити їх відповідність вимогам;

- виконати експериментальне дослідження результативності платформи та проаналізувати отримані показники;

- сформулювати висновки та рекомендації щодо можливостей подальшого вдосконалення системи.

Наукова новизна роботи полягає у розробці архітектурного рішення та практичної реалізації інтерактивної освітньої веб-платформи з використанням сучасного технологічного стеку, що поєднує зручність користування, масштабованість і адаптивний дизайн. Запропонована система відрізняється від аналогів орієнтацією на гнучкість, поєднанням форматів, ефективною інтеграцією найсучасніших інтерактивних та можливістю швидкої модифікації під конкретні навчальні програми.

Практична цінність роботи полягає у створенні веб-платформи, яку можна використовувати для онлайн мовних шкіл та організації онлайн-курсів не лише з іспанської, а й інших мов. Розроблена система спрощує процес запису на навчання, забезпечує ефективну взаємодію між викладачами та студентами, а також сприяє підвищенню цифрової присутності навчального закладу або приватного викладача. Отримані результати можуть бути впроваджені в діяльність мовних шкіл, освітніх стартапів і навчальних центрів або використані як основа для подальших досліджень і вдосконалення технологій у сфері онлайн-освіти.

У роботі застосовуються методи об'єктно-орієнтованого моделювання, UML-діаграмування, а також підхід Rational Unified Process (RUP) для опису специфікацій вимог.

Структура роботи складається з трьох розділів. Перший розділ містить аналіз предметної області, стану онлайн-освіти, особливостей викладання іспанської мови та огляд мовних платформ і методів розробки веб-систем. Другий розділ присвячений проєктуванню та реалізації веб-платформи, включаючи вибір методології, UML-моделювання, обґрунтування технологій і розробку функціональних компонентів. Третій розділ охоплює експериментальне дослідження роботи системи та аналіз отриманих

результатів. Робота поєднує теоретичний аналіз і практичну реалізацію веб-платформи для онлайн-навчання іспанської мови.

Апробація результатів дослідження була здійснена шляхом публікації наукової статті у «Науковому віснику Луцького національного технічного університету».

# РОЗДІЛ 1

## АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ, МЕТОДІВ РОЗРОБКИ ТА ПОСТАНОВКА ЗАВДАННЯ

### 1.1 Стан онлайн-освіти в Україні та світі

Сфера освіти зазнала глобальної трансформації під впливом цифрових технологій. Онлайн-освіта – це форма дистанційного навчання, що передбачає використання цифрових технологій та інтернету для передачі знань [1]. Враховуючи освітнє призначення даної роботи, варто зазначити що онлайн-курси є ключовою складовою онлайн-освіти. Упродовж останніх років, така форма навчання стала невід’ємною частиною освітнього простору в усьому світі.

Онлайн-курс представляє систематизовану навчальну програму, яка проводиться через цифрові платформи та Інтернет. Він передбачає послідовне вивчення матеріалу за допомогою відеолекцій, текстових ресурсів та практичних завдань. Відповідно, така організація навчального процесу дозволяє здобувати знання незалежно від місця та часу, забезпечуючи гнучкість для студентів.

Особливе місце у розвитку онлайн-освіти займають мовні платформи. Для детального вивчення галузі було проведено дослідження сучасного ринку освіти в контексті надання курсів. Проведений аналіз дозволив виявити, що онлайн-освіта охоплює широкий спектр форматів та надвисоку динаміку зростання [2-3]. Серед основних форматів доцільно виділити індивідуальні, масові, відкриті онлайн-курсів та спеціалізовані платформи для їх надання. За офіційними даними міжнародної аналітичної компанії Grand View Research, в одному з аналітичних звітів йдеться, що розмір глобального ринку освітніх технологій у 2024 році становив близько 163,5 млрд доларів, і до 2030 року він може зрости до 348,4 млрд доларів [3]. Ці дані свідчать про сталу тенденцію до зростання попиту на досліджувану дистанційну форму навчання.

Аналізуючи міжнародний ринок, доцільно зазначити, що навчальні платформи Coursera, Udemy, edX та Duolingo, об'єднують мільйони користувачів, пропонуючи різноманітні курси [4-7]. Провідні університети, зокрема Гарвард, Стенфорд, Оксфорд, активно інтегрують, онлайн-навчання, онлайн-курси у свої навчальні програми, що є очевидним кроком зважаючи на популяризацію та ефективність даного формату. Відповідно існують підстави вважати, що рівень довіри до цього формату надзвичайно високий. З кожним днем все більше людей вибирають онлайн-формат як головну форму, заміну або додаток до класичної освіти .

В Україні цифрова трансформація освіти стала справжнім порятунком в період пандемії COVID-19. Зокрема, в даному випадку доцільно згадати події останніх років, коли під впливом масового карантину майже кожен навчальний заклад України звернулися до цифрових технологій в пошуках рішення. Онлайн-формат став не просто альтернативою, а єдиною можливою формою навчання в багатьох регіонах країни. Таким чи, це спричинило зростання кількості державних і приватних ініціатив, орієнтованих на створення якісних цифрових освітніх ресурсів. Водночас, в зв'язку з подіями останніх років, зокрема повномасштабного вторгнення, який значною мірою вплинув на демографічну ситуацію в країні, виникли масові міграції серед громадян [8]. Серед мільйонів людей, які вимушено покинули свій дім та переїхали до нової країни, виникла потреба у вивченні іноземної мови, що в свою чергу спричинило різке збільшення попиту на спеціалізовані для цього ресурси.

За даними соціологічних досліджень Київського міжнародного інституту соціології, абсолютна більшість українців, а саме 99 %, вважають що вивчення мов є необхідним для кожного громадянина [9]. Автори підкреслюють значимість та низку вагомих факторів даного напрямку, що все-таки демонструє актуальність досліджуваної ніші.

Серед найпоширеніших мов для вивчення іспанська посідає одне з провідних місць, оскільки належить до найпопулярніших мов у світі. Чимало людей прагнуть її вивчати для мандрівок, праці чи здобуття освіти. В Україні

відчувається брак адаптованих онлайн-ресурсів, які б об'єднували живі уроки, інтерактивні вправи та сучасний підхід до викладання. Тому створення нової платформи – це відгук на справжню потребу, яка є серед українських користувачів сьогодні.

Таким чином, проведений аналіз сучасного стану онлайн-освіти у світі та в Україні дозволяє зробити кілька узагальнень. По-перше, онлайн-освіта перестала бути альтернативою традиційній і стала самостійною формою навчання глобально розвивається. По-друге, Україна активно долучається до цього процесу, демонструючи потенціал у створенні власних інноваційних освітніх продуктів. По-третє, порівняння зарубіжного та українського досвіду доводить, що провідні університети активно інтегрують онлайн-навчання у свої навчальні програми. З глобальної та локальної точок зору стає очевидним підсумок, що онлайн-освіта вже не є тимчасовим трендом, а стала невід'ємною формою отримання знань.

## **1.2 Особливості викладання іспанської мови онлайн**

Акцентуючи увагу на об'єкті дослідження варто зазначити, що іспанська понад 600 мільйонів людей, з яких близько 460-493 мільйонів є носіями. Вона є офіційною в 21 країні, входить до п'ятірки найуживаніших мов світу та стабільно залишається однією з найпопулярніших для вивчення як іноземна. У глобалізованому світі знання іспанської відкриває широкі можливості для навчання, працевлаштування та культурного обміну, а тому попит на якісне викладання цієї мови постійно зростає [10].

У межах дослідження важливо розглянути методичні підходи до викладання іспанської мови онлайн. У лінгводидактиці традиційно виокремлюють три ключові методи. Перший – граматико-перекладний, що робить акцент на правилах та перекладі текстів. Другий – аудіолінгвальний, який передбачає багаторазове повторення мовних структур. Третій – комунікативний, орієнтований на формування умінь спілкування в реальних

ситуаціях [11]. Для онлайн-формату саме комунікативний підхід є найбільш ефективним, адже поєднує теоретичний матеріал з активною практикою: студент не лише засвоює граматичні конструкції, а й одразу застосовує їх у мовленні.

Викладання іноземної мови онлайн має свої особливості, переваги та недоліки, які суттєво впливають на структуру занять, та взаємодію між викладачем і студентом. Іспанська мова має певні фонетичні, граматичні та лексичні особливості, які вимагають спеціальних підходів у навчанні. Зважаючи на багатогранність та варіативність даної мови, існує низка нюансів, які особливо важливі для опанування в процесі навчання:

- правильна вимова;
- наголоси;
- інтонація;
- усне мовлення та практика вживання лексики в контексті.
- складні граматичні конструкції;
- різноманіття часів;
- особливості лексики.

Фахівці з викладання іспанської відзначають специфіку цієї романської мови, що потребує особливого методичного підходу. Дослідники П. Гарсія та М. Лопес у роботі «Enseñanza práctica del español» наголошують на критичній ролі фонетичних вправ на початкових етапах навчання, адже іспанська вимова має чіткі правила читання, які студенти мають засвоїти з перших занять [12]. Автори підкреслюють, що відпрацювання наголосів та інтонаційних моделей через аудіовізуальні засоби суттєво прискорює формування правильних артикуляційних навичок. Такий підхід особливо актуальний для дистанційного формату, де викладач може використовувати записи носіїв мови та інтерактивні вправи для закріплення матеріалу. Правильна вимова, акцент та точна артикуляція звуків становлять основу ефективного опанування досліджуваної мови. Досвід провідних методистів підтверджує, що поєднання фонетичних

тренувань із регулярними короткими діалогами сприяє автоматизації правильної вимови.

Онлайн-платформа для надання курсів за своїм призначенням прирівнюється до школи через її освітню функцію, тому в такому разі доцільно провести їх порівняння [13]. Для систематизації основних відмінностей між традиційним і дистанційним форматами нижче подано таблицю 1.1.

Таблиця 1.1 – Порівняння онлайн та фізичних шкіл вивчення іспанської мови

Критерій дослідження	Характеристичне порівняння форматів	
	Онлайн-школа	
Зручність	Можна навчатися з будь-якого місця	Зручність
Доступність для регіонів	Доступно у будь-якій точці світу	Доступність для регіонів
Живе спілкування	Є, але через екран (Zoom тощо)	Живе спілкування
Ціна	Зазвичай дешевше	Ціна
Дисципліна	Потрібна самоорганізація	Дисципліна
Якість матеріалів	Часто інтерактивні, сучасні, з відео, вправами онлайн	Якість матеріалів
Технічні вимоги	Потрібен стабільний інтернет і пристрій	Технічні вимоги
Соціальне середовище	Менше живого спілкування з одногрупниками	Соціальне середовище

Як бачимо з таблиці, реалізоване порівняння демонструє, що онлайн-викладання має низку переваг, серед яких основними є гнучкість, економічність і можливість повторного доступу до матеріалів. Водночас важливо враховувати психологічний аспект – зниження рівня соціальної взаємодії може впливати на мотивацію студентів. Тому ефективне викладання іспанської мови онлайн

передбачає поєднання педагогічної майстерності, цифрової грамотності викладача та використання інноваційних технологій

В контексті даної наукової роботи передбачається, що надання курсів іспанської мови відбуватиметься у форматі інтерактивного онлайн-навчання із залученням сучасних зовнішніх ресурсів, зокрема інструмент Miro для візуальної взаємодії та Zoom для проведення живих онлайн-уроків. Окремо зазначається, що даний організація курсів на сайті реалізована таким чином, що студент має змогу проходити самостійно та за участю кваліфікованого викладачів, зважаючи на наявність великої кількості мультимедійних інтерактивних, граматичних та лексичних матеріалів. Ця комбінація забезпечує живе спілкування з викладачем і можливість показувати складну граматику візуально, що дуже важливо.

Miro – це інтерактивний інструмент для візуалізації, який вирішує проблему подачі матеріал структуровано онлайн. На цій платформі дозволяє створювати інтерактивні дошки, де є граматичні таблиці, лексичні карти, діалоги та вправи. Студенти працюють з цим синхронно, тобто переміщують елементи, заповнюють пропуски, складають речення з готових частин. Якщо використовувати різні кольори для граматичних категорій наприклад для часів дієслова це допомагає краще запам'ятати систему. Коли матеріал організований візуально на дошці викладач може адаптувати пояснення прямо під час заняття в залежності від того як студенти розуміють тему.

Zoom забезпечує стабільний відеозв'язок для індивідуальних, парних і для групових занять, додатково дозволяється записувати заняття. Також, серед переваг є демонстрація екрану, що допомагає пояснювати складні граматичні правила та можна організувати роботу в малих групах щоб студенти практикували діалоги між собою. Запис заняття є великою перевагою, оскільки студент може переглянути матеріал ще раз для кращого засвоєння, проте в звичайній школі такого немає. Якщо порівнювати зі Meet то Zoom стабільніший і має більше інструментів для інтерактиву.

В межах даної роботи передбачається надання онлайн-курсів іспанської на чотирьох рівнях A1 – це початковий рівень, A2 – елементарний, B1 – середній та B2 – вище середнього. Курси були розроблені згідно усіх вимог освітньої системи з реалізацією комунікативного підходу. Кожен рівень це 48 уроків, тобто стільки ж академічних годин, оскільки одне триває 60 хвилин кожне. Курси в контексті у випадку з проходженням з викладачем є в трьох форматах, зокрема індивідуальні, парні або в групі до 4 студентів. Окремо зазначається, що студент також може самостійно проходити курс, оскільки формат та інтерактивні матеріали уможливають таку форму навчання. Студент реєструється на курс, заповнюючи заявку, де вказує який рівень вивчатиме та за бажанням проходить тест в Google Form щоб визначити свій рівень. Адміністратор обробляє заявку зв'язується через інтегровані месенджери для узгодження часу і дати заняття та відправляє посилання на безкоштовне пробне заняття в Zoom. Після пробного заняття студент отримує і доступ до матеріалів на Miro. Кожне заняття можливо записувати, тому його зручно переглядати згодом.

Окремо зазначається, що живий формат викладання курсів здійснюється за участю кваліфікованих викладачів, що дозволяє забезпечити високий рівень засвоєння матеріалу та індивідуальний підхід до кожного студента. Такий підхід орієнтований на створення комфортного навчального середовища, де студенти можуть здобувати знання з будь-якої точки світу в зручній для них час. Проведені дослідження свідчать, що успішність засвоєння матеріалу в онлайн-форматі залежить від рівня інтерактивності занять, наявності аудіовізуальної підтримки, частоти зворотного зв'язку й використання мультимедійних матеріалів [15]. Таким чином, учні матимуть змогу ознайомитися з повною інформацією про доступні курси, формати та умови навчання, вартість занять з викладачем, а також детальніше дізнатися свій рівень мови.

Узагальнюючи, можна стверджувати, що онлайн-навчання іспанської мови є ефективним лише за умови комплексного підходу – поєднання

методичної бази, технічного забезпечення та мотиваційної підтримки студентів. Важливо забезпечити на платформі можливість інтерактивної взаємодії, підтримку мультимедійного контенту та просту комунікацію між учнями і викладачами. Лише за таких умов дистанційне навчання може бути дійсно ефективним і привабливим для учня.

### **1.3 Аналіз існуючих мовних платформ для надання курсів з іспанської та формування вимог до веб-платформи**

Онлайн-платформи для навчання стали популярним напрямом освітнього бізнесу в Україні. Вони пропонують інтерактивні уроки, відеолекції, індивідуальні консультації, що дає можливість навчатися без обмежень у просторі й часі.

Проведений аналіз українських мовних платформ показав, що ринок залишається відносно відкритим для створення інноваційних рішень у сфері вивчення іноземних мов, зокрема іспанської.

На комерційному ринку українських веб технологій вже існує певна кількість розробок за необхідною нам спеціалізацією. У контексті даної роботи, відповідно до призначення, було розглянуто низку спеціалізованих платформ надання курсів з іспанської, проте більш детально досліджено наступні: Evermore, Espanita, Memrise та Alfa Grant [14-17].

Аналіз ринку мовних платформ засвідчив наявність потенціалу для впровадження інноваційних рішень у сфері викладання іспанської мови. Для систематизованого порівняння існуючих програмних рішень було обрано ключові критерії оцінювання: формат навчання, швидкість завантаження платформи, наявність живих занять з викладачем, можливість пробного заняття та інтеграція інтерактивних інструментів. Результати порівняльного аналізу подано у таблиці 1.2.

Таблиця 1.2 – Порівняльний аналіз наявних рішень

Існуючі платформи аналоги	Критерії дослідження				
	Формат навчання	Швидкість завантаження платформи (сек)	Живі заняття з викладачем	Пробне заняття	Інтеграція інтерактивних інструментів
Evermore	Групові, індивідуальні, парні	1.5	Так	Так, безкоштовне	Так
Espanita	Групові та індивідуальні	1.9	Ні	Так	Так
Memrise	Самостійне	2.8	Ні	Ні	Ні
Alfa Grant	Тільки індивідуальні	1.2	Так	Так, безкоштовне	Так

Як бачимо з таблиці, Espanita та Alfa Grant займають оптимальну позицію на ринку освітніх технологій. Обидві платформи пропонують живі заняття з викладачами, пробні заняття та інтеграцію інтерактивних інструментів, що забезпечує комплексний підхід до навчання. Водночас Alfa Grant, фокусується виключно на індивідуальних заняттях з безкоштовним пробним уроком, забезпечує персоналізований підхід до кожного студента. Така комбінація якісної взаємодії з викладачем, можливості апробації сервісу без фінансових ризиків та доступу до сучасних навчальних інструментів робить нашу платформу необхідним та ефективним рішенням для студентів, які прагнуть досягти високих результатів у вивченні мови.

Враховуючи результати аналізу існуючих рішень, специфіку викладання іспанської мови онлайн та сучасні тенденції у сфері освітніх технологій, було сформовано комплекс вимог до проєктованої веб-платформи. Розробка веб-платформи для мовної школи передбачає створення сучасного ресурсу, що виконуватиме низку функцій та відповідатиме певним характерним вимогам.

Основна мета платформи – зробити процес навчання максимально гнучким, інтерактивним та доступним з будь-якої точки світу, забезпечуючи при цьому якісну взаємодію між студентами та викладачами.

Розробка веб-платформи для вивчення іспанської мови передбачає створення сучасного ресурсу, що виконуватиме низку функцій, та відповідатиме певним характерним вимогам. Важливо, щоб поєднувалась інформативність, зручність навігації, сучасний адаптивний дизайн і можливість взаємодії з користувачами. Проведений аналіз існуючих аналогів показав, що успішні освітні веб-платформи поєднують інформативність, інтуїтивну навігацію, адаптивний дизайн і гнучку систему управління контентом.

Зважаючи на специфіку освітньої тематики головними завданнями такої платформи є презентація курсів, умов навчання та забезпечення можливості зв'язку з адміністрацією школи.

Структура сайту повинна бути логічно побудованою та зрозумілою для користувача. На першому рівні навігації мають бути представлені такі ключові модулі: головна сторінка, загальна інформація, сторінки з описом курсів, блок «Кому підходить», розділ з видами занять і цінами, приклади навчальних матеріалів, форма для запису на пробне заняття та контакти. Кожна сторінка повинна мати чітку структуру з заголовками, текстовим наповненням, зображеннями, кнопками дій та інтерактивними елементами. Важливо також реалізувати футер із контактною інформацією, посиланнями на месенджери та логотипом школи.

З функціональної точки зору платформа має забезпечити наступні можливості:

- відображення структурованої інформації про курси (рівні A1, A2, B1, B2), тривалість навчання, кількість занять, формат проведення та умови участі;
- доступ до тесту на визначення рівня іспанської мови (через інтеграцію Google Form);
- можливість переглянути приклади навчального матеріалу;

- підтримка адаптивності для перегляду на мобільних телефонах, планшетах і десктопах;
- реалізація інтерактивної форми для запису на пробне заняття з обробкою введених даних ;
- автоматичне збереження заявки у базі даних та відправка підтвердження адміністратору;
- швидке зв'язування з представниками школи через кнопки месенджерів;
- можливість проходження тесту на визначення рівня мови;
- можливість проходити курси інтерактивно та гнучко.

Окрім користувацьких функцій, важливо забезпечити ефективну адміністративну панель, яка дозволяє:

- додавати, редагувати та архівувати курси;
- управляти базою студентів і заявок;
- відстежувати активність користувачів.

Крім того, сайт має бути простим у підтримці та масштабуванні. Це передбачає реалізацію структури на базі компонентного підходу, при якому кожен блок реалізований як окремий модуль, що дозволяє у майбутньому розширювати функціональність без суттєвого переписування вже наявного коду.

Нефункціональні вимоги спрямовані на підвищення зручності використання веб-платформи. З естетичної точки зору дизайн має відповідати сучасним трендам UI/UX: мінімалізм, чітка типографіка, інтуїтивна структура сторінок, дотримання одного стилю дизайну, використання кольорової гами, що асоціюється надійністю та професіоналізмом [18]. Інтерфейс повинен забезпечувати швидку навігацію, адаптивність і приємне візуальне сприйняття. Це сприяє підвищенню рівня довіри користувачів до школи та покращує конверсію відвідувань у реєстрації.

Таким чином, вимоги до структури платформи охоплюють як естетичні й інформаційні аспекти, так і технічні – взаємодія з базою даних, обробка форм,

динамічний контент та адаптивність. Врахування цих вимог є критично важливим для забезпечення ефективності платформи як інструменту залучення і взаємодії з потенційними учнями.

#### **1.4 Огляд і аналіз методів та засобів розробки веб-платформ**

Розробка сучасних веб-платформ є комплексним процесом, що потребує вибору відповідних технологій, архітектурних підходів та інструментів програмування. На сьогоднішній день існує велика кількість мов програмування, спрямованих на розробку сучасних веб-рішень, зокрема платформ дистанційного навчання. На основі аналізу наукових джерел, підручників та фахової літератури було здійснено огляд основних мов та засобів, які можуть бути застосовані під час створення веб-платформи для онлайн-навчання.

Питання структуризації веб-розробки у своїй праці досліджував С. В. Баран у навчальному посібнику «Основи WEB-програмування». Автор підкреслює важливість поетапної організації веб-проектів, починаючи від аналізу вимог і закінчуючи тестуванням та розгортанням застосунку [19]. Окрему увагу він приділяє необхідності розмежування клієнтської та серверної логіки, що забезпечує масштабованість та керованість веб-систем. У роботах Д. Фланагана детально розглянуто концепції JavaScript та практики створення інтерактивних клієнтських інтерфейсів. У праці Н. Захарченко «Веб-програмування» [20] зазначається, що вибір технології насамперед залежить від типу задачі: контент-орієнтовані системи, інтерактивні сервіси або навчальні платформи потребують різних інструментів. На основі цих джерел було виконано аналіз можливих засобів реалізації.

PHP – серверна мова програмування, призначена для формування динамічного веб-контенту. Вона належить до скриптових мов, орієнтованих на роботу з HTML-документами, обробку HTTP-запитів та взаємодію з веб-сервером. PHP залишається однією з найпоширеніших мов серверної розробки.

Як зазначає О. Васильєв у посібнику «Програмування мовою PHP», PHP й надалі використовується як базова серверна мова для створення динамічних веб-сторінок та забезпечує підтримку популярних фреймворків, серед яких Laravel і Symfony [21]. Перевагою PHP є простота освоєння та розвинена спільнота, що дозволяє швидко розробляти й розгортати типові веб-сервіси. PHP доцільно розглядати для платформ, де основний акцент робиться на роботі з контентом, автоматизованих панелях управління та серверних шаблонах.

Наступним поширеним інструментом розробки є Python – високорівнева мова програмування з динамічною типізацією та універсальним застосуванням. Вона відома лаконічним синтаксисом, високою читабельністю та широким набором бібліотек.

Python використовується як у веб-розробці, так і в аналітиці даних, машинному навчанні та автоматизації. Водночас він відомий своїм простим синтаксисом та високою читабельністю коду, що сприяє швидкій командній розробці.

Для веб-додатків найбільш поширеними фреймворками є Django та Flask. Як зазначає А. О. Костюченко у навчальному посібнику «Основи програмування мовою Python», Django забезпечує високий рівень безпеки, швидкість прототипування та чітку структуру проєктів, що робить його доцільним вибором для освітніх платформ і сервісів із великою кількістю логіки [22]. Python також є оптимальним інструментом для інтеграції адаптивних рекомендаційних систем, аналітики та оцінювання навчальних результатів за допомогою моделей машинного навчання.

Ще однією з найпоширеніших мов є JavaScript – стандарт клієнтської веб-розробки. JavaScript є динамічною інтерпретованою мовою, що виконується у браузері та забезпечує інтерактивність інтерфейсу. У роботі Д. Фланагана відзначається, що JavaScript дає високу інтерактивність веб-інтерфейсів [23]. Поява Node.js дозволила використовувати JavaScript і на серверному боці, що значно розширило його можливості.

JavaScript демонструє високу динаміку розвитку. Сьогодні ця мова програмування використовується для створення різноманітних додатків, від ігор до складних веб-сервісів. Перевагою JavaScript є можливість створення повного стеку технологій (frontend + backend), що спрощує розробку та зменшує витрати на підтримку. Для візуалізації та систематизації дослідження, результати подано у таблицю 1.3.

Таблиця 1.3 – Порівняльний аналіз засобів розробки веб-платформ

Характеристика	Мови програмування		
	PHP		PHP
Частка використання на ринку (частка, %)	20 %	Частка використання на ринку (частка, %)	20 %
Продуктивність	Середня	Продуктивність	Середня
Підтримка інтерактивності	Низька без додаткових засобів	Підтримка інтерактивності	Низька без додаткових засобів
Масштабованість	Обмежена без оптимізації	Масштабованість	Обмежена без оптимізації
Кількість розробників (млн осіб)	6.5	10.1	19.9
Наявність фреймворків	Велика кількість	Django/Flask	React, Vue, Angular, Express
Середня річна зарплата	68000 \$	68000 \$	68000 \$
Сфера доцільного застосування	CMS, корпоративні сайти	Аналітика, освітні системи	Навчальні платформи, реальний час
Підтримка API	Середня	Висока	Дуже висока

Аналіз літературних джерел та технологічних засобів показав, що кожна з розглянутих мов програмування має свої особливості та переваги. PHP підходить для класичних веб-платформ із переважно серверною логікою. Python є ефективним для проєктів, що потребують складної внутрішньої

аналітики або інтелектуальних компонентів. JavaScript у поєднанні з Node.js та сучасними фронтенд-фреймворками забезпечує високу продуктивність та інтерактивність, що є ключовими вимогами для сучасних систем онлайн-навчання.

Таким чином, для розробки платформи для надання онлайн-курсів з іспанської мови було обрано стек JavaScript-технологій. Основою серверної частини став Node.js, що забезпечує швидку обробку запитів та масштабованість застосунку. Такий вибір технологій забезпечує оптимальне поєднання продуктивності, гнучкості та можливості подальшого масштабування платформи. Для клієнтського інтерфейсу використано React, який дозволив створити динамічний, інтерактивний та зручний для користувача веб-інтерфейс.

### **1.5 Постановка завдання на кваліфікаційну роботу магістра**

На основі проведеного в попередній підрозділах аналізу стану онлайн-освіти, особливостей викладання іспанської мови у та визначених вимог до структури і функціоналу платформи, було сформовано перелік завдань, які необхідно реалізувати у межах магістерської роботи. Ці завдання спрямовані на створення сучасної веб-платформи, для надання курсів з іспанської мови.

У межах роботи передбачається виконати такі основні завдання:

- проаналізувати наявні аспекти онлайн-навчання, а також наявні технічні рішення в контексті мовних курсів;
- визначити та формалізувати вимоги до веб-платформи, включаючи структуру, основні функціональні можливості та критерії якості;
- обґрунтувати вибір технологічного стеку та архітектурних рішень, придатних для реалізації платформи мовної школи;
- розробити UML-моделі, що відображають логіку роботи системи, взаємодію користувачів та структуру даних;

- реалізувати веб-платформу відповідно до визначених вимог, включаючи модулі презентації курсів, визначення рівня, запису на заняття та зворотного зв'язку;
- інтегрувати можливість проходження курсів;
- провести тестування функціональних компонентів та оцінити їх відповідність вимогам;
- виконати експериментальне дослідження результативності платформи та проаналізувати отримані показники;
- сформулювати висновки та рекомендації щодо можливостей подальшого вдосконалення системи.

Досягнення поставлених цілей забезпечить створення повнофункціональної веб-платформи для організації онлайн-курсів з іспанської мови. Результати дослідження матимуть практичну цінність для мовних шкіл, освітніх стартапів та викладачів, які прагнуть автоматизувати процес залучення студентів та підвищити цифрову присутність у середовищі онлайн-освіти.

## РОЗДІЛ 2

# ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВЕБ-ПЛАТФОРМИ З НАДАННЯ КУРСІВ ІСПАНСЬКОЇ МОВИ

### 2.1 Вибір методології та інструментів проєктування

Для планування, проєктування та розробки платформи було обрано методологію Rational Unified Process. RUP належить до гнучких і водночас структурованих підходів у сфері створення програмного забезпечення. Ця методологія поєднує системність традиційних методів і адаптивність сучасних Agile-практик, добре підходить для проєктів, що потребують поетапної реалізації, чіткого розуміння вимог та можливості поступового вдосконалення.

RUP базується на об'єктно-орієнтованому підході та підтримує використання UML-діаграм (Unified Modeling Language) для візуалізації різних аспектів системи. Однією з ключових особливостей методології є її ітераційна природа. Таким чином, розробка відбувається не одним великим етапом, а у вигляді циклів, під час яких створюється частина функціональності, яка поступово розширюється й удосконалюється. Ітераційний характер RUP дозволяє поступово розробляти функціональність продукту, перевіряти її на кожному етапі, коригувати вимоги та вдосконалювати архітектуру без порушення загальної логіки [24].

Методологія RUP складається з чотирьох основних фаз:

– фаза початку (Inception) – визначаються цілі проєкту, його основна ідея, описується концепція та попередня оцінка ресурсів. У контексті даної платформи це етап формування бачення системи, визначення основних функцій, структури та цільової аудиторії;

– фаза уточнення (Elaboration) – деталізуються функціональні й нефункціональні вимоги, створюється архітектура майбутньої системи, проєктуються основні компоненти. На цьому етапі розроблено макети інтерфейсу, діаграми варіантів використання, визначено технології та стек розробки;

– фаза конструювання (Construction) – відбувається безпосередня розробка функціоналу, створення коду, підключення бази даних, реалізація форм, інтеграція з месенджерами, тестування окремих частин системи. У даному випадку це створення адаптивного інтерфейсу з усіма запланованими сторінками;

– фаза впровадження (Transition) – система готується до розгортання та подальшого використання. Сюди входить розміщення платформи на хостингу, тестування всієї функціональності в реальних умовах, адаптація під мобільні пристрої.

Основними принципами методології є:

– ітераційність – кожна фаза завершується створенням працездатної версії продукту, що знижує ризики та підвищує якість;

– керування вимогами – усі вимоги документуються та відстежуються протягом життєвого циклу;

– візуальне моделювання – система проектується за допомогою UML-діаграм, що підвищує розуміння архітектури;

– компонентність – застосунок розробляється у вигляді незалежних модулів, які легко модифікувати або масштабувати.

Важливою перевагою RUP є можливість ефективно контролювати ризики, поступово перевіряючи кожен крок і вносячи зміни на ранніх етапах. Такий підхід особливо зручний для навчального проєкту, оскільки дозволяє поступово реалізовувати всі компоненти та коригувати рішення у процесі роботи.

Використання Rational Unified Process дозволяє зберегти баланс між плануванням і гнучкістю, що є оптимальним для системи на кшталт інформаційної платформи для вивчення іноземної мови. Така концепція допомогла систематизувати етапи розробки, уникнути хаотичних змін і досягти логічної, поетапної реалізації кожного функціонального блоку.

У межах методології RUP формалізація ролей є невід’ємною частиною процесу моделювання, оскільки визначає учасників системи, їхні обов’язки та

взаємодію з артефактами, забезпечуючи чітку структуру доступу й логіку виконання сценаріїв використання. Визначення ролей дозволяє чітко структурувати логіку доступу, обов'язки, функції та обмеження для кожної категорії користувачів. У межах розроблюваної системи було виділено дві основні ролі: адміністратор і учень.

Роль «Адміністратор». Адміністратор – це користувач, який відповідає за загальне керування платформою. Його права доступу та роль є найрозширенішими і включають повний доступ до всіх функцій сайту, а також можливість вносити зміни в його структуру й наповнення.

Основні функції адміністратора:

- перегляд та обробка заявок від потенційних учнів;
- редагування або оновлення інформації про курси та ціни;
- керування візуальним контентом сайту;
- за потреби – модерація контенту, що подається викладачами.

Мета ролі адміністратора – забезпечити цілісність, актуальність і функціональність платформи. Його діяльність формує бекенд-інфраструктуру, а від ефективності виконання завдань залежить стабільність роботи системи.

Роль «Учень». Учень – це основний кінцевий користувач платформи. Саме для нього реалізована більшість функціоналу інтерфейсу. Його взаємодія із системою зосереджена на інтерфейсі користувача, який надає зручні способи отримання інформації та запису на курси. Роль учня не передбачає авторизації чи створення особистого кабінету, проте дозволяє йому ознайомитися з навчальними можливостями школи, швидко здійснити запис і розпочати навчання.

Основні можливості учня:

- перегляд усієї публічної інформації на сайті (про школу, курси, викладачів, формати занять);
- ознайомлення з цінами, умовами навчання;
- проходження онлайн-тесту на визначення рівня володіння іспанською;

- перегляд прикладів навчальних матеріалів (у вигляді відео, зображень або посилань на інтерактивну дошку Miro);
- натискання на кнопки для зв'язку зі школою через Telegram, Instagram або через електронну пошту;
- заповнення заявки на безкоштовне пробне заняття через спеціальну форму.

Учень не має доступу до адміністративних функцій і не виконує дій, які змінюють базові параметри системи. Його роль спрямована на отримання інформації, комунікацію з адміністрацією та ініціацію навчального процесу.

Таким чином, визначення ролей користувачів забезпечує чітку організацію доступу до функціоналу системи, спрощує навігацію інтерфейсом та створює базу для майбутнього розширення можливостей платформи відповідно до потреб різних категорій користувачів.

Доцільно розглянути як виглядає опис успішного сценарію та альтернативних ситуацій для веб-платформи в стилі RUP. Основний успішний сценарій запису користувача на курс реалізовано як послідовність взаємодій між користувачем, клієнтською частиною системи та сервером. Спочатку користувач переходить на веб-сайт платформи Viva Lingua. У відповідь серверна частина застосунку завантажує та відображає головну сторінку, що містить інформаційні банери, меню навігації та перелік доступних курсів різних рівнів.

Переглянувши доступні пропозиції, користувач обирає потрібний курс, наприклад рівень A1, та натискає кнопку «Записатись». Після цього на екрані відкривається модальне вікно з формою подачі заявки. Користувач заповнює обов'язкові поля: рівень, номер телефону, нікнейм у соціальних мережах (Instagram або Telegram) та за потреби залишає коментар. Після завершення заповнення форми він натискає кнопку «Надіслати».

На цьому етапі клієнтська частина застосунку, реалізована на основі React, формує запит та надсилає його на сервер за допомогою бібліотеки Axios. Сервер, побудований на Node.js та Express, обробляє отримані дані й зберігає

заявку в хмарній базі даних MongoDB Atlas. Після успішного занесення інформації до бази даних адміністратор платформи автоматично отримує сповіщення про нову заявку через інтегровану адміністративну панель. Завершальним етапом є відображення на екрані користувача повідомлення про успішну подачу заявки: «Дякуємо! Заявка прийнята».

Разом із основним сценарієм передбачено кілька альтернативних, що моделюють можливі помилки або виняткові ситуації. Якщо сервер тимчасово недоступний, користувач не може завантажити сайт і бачить відповідне повідомлення про недоступність ресурсу. За відсутності інтернет-з'єднання сторінка не завантажується, а браузер інформує про помилку мережі.

Якщо у формі не заповнені обов'язкові поля, система блокує відправлення заявки і відображає повідомлення з вимогою заповнити необхідні дані. У випадку помилки під час надсилання форми – наприклад, через збій на сервері – користувач отримує відповідне повідомлення з пропозицією повторити спробу пізніше. Окремо опрацьовується ситуація некоректного формату введених даних: у разі введення неправильного номера телефону або іншої невідповідної інформації система виводить підказку щодо правильного формату.

Формалізація ролей користувачів є основою безпечної й логічної архітектури системи. Визначення чітких ролей – адміністратора та учня – дозволяє впорядкувати взаємодію користувачів із системою, підвищити рівень безпеки та забезпечити прозорість функціональних зв'язків між компонентами.

Розмежування прав доступу сприяє зниженню ризику помилок і забезпечує цілісність даних, що особливо важливо для освітніх платформ, де одночасно обробляється значний обсяг персональної інформації. Такий підхід створює гнучку архітектуру, придатну до масштабування та подальшого розширення системи – наприклад, інтеграції нових ролей.

Таким чином, формалізація користувацьких ролей не лише підвищує ефективність управління платформою, але й забезпечує технологічну

стабільність та надійність роботи системи в умовах розширення функціоналу й зростання кількості користувачів.

Використання методології RUP у межах даної кваліфікаційної роботи забезпечило чітку структурованість і контроль над усіма етапами життєвого циклу системи, дозволивши поєднати планування, проектування, розробку та тестування в єдиній логічній послідовності. Ітераційний підхід забезпечив гнучке реагування на зміни, а наявність артефактів підвищила прозорість і наукову обґрунтованість процесу. Важливою складовою RUP стала формалізація користувацьких ролей, що визначає учасників системи та їхні функції. Це підвищило ефективність управління платформою, забезпечило технологічну стабільність і створило основу для подальшого масштабування системи із зростанням кількості користувачів та розширенням функціоналу.

## **2.2 UML-моделювання системи**

Одним із важливих етапів проектування програмного забезпечення є візуалізація його логіки, структури та поведінки через графічні моделі. Для цього використовуються UML-діаграми, які дають змогу наочно подати основні компоненти системи та зв'язки між ними. Unified Modeling Language – це мова візуального моделювання, яка дозволяє графічно описати структуру та поведінку програмної системи [25]. Використання UML діаграм дає змогу систематизувати вимоги, спланувати архітектуру проєкту, виявити ключові сутності, їхні зв'язки та взаємодію користувачів із системою. Завдяки цьому можна заздалегідь продумати логіку роботи програмного забезпечення та уникнути помилок у процесі реалізації. Такий підхід спрощує розуміння внутрішньої організації платформи й полегшує як реалізацію, так і подальшу підтримку.

Використання UML сприяє зменшенню ризиків невідповідності між очікуваним функціоналом і кінцевим результатом, підвищує передбачуваність розробки та полегшує комунікацію між сторонами. Щоб ефективно реалізувати

цей етап, було необхідно обрати інструмент, який дозволяє швидко, зручно та якісно створювати UML-діаграми відповідно до поставлених завдань [26]. Для обґрунтованого вибору було проведено порівняння трьох популярних платформ: StarUML, Lucidchart та diagrams.net. Доцільно розглянути характеристики кожної з платформ.

StarUML – це професійне програмне забезпечення для створення діаграм, яке надає широкі можливості з моделювання систем на основі UML. Воно підтримує повний набір діаграм і дозволяє налаштовувати структуру проєкту відповідно до стандартів RUP. Проте ця програма потребує встановлення на комп'ютер і має обмежений безкоштовний доступ, що ускладнює її використання в навчальному контексті.

Diagrams.net – зручний веб-інструмент, який дозволяє створювати прості схеми, графи й діаграми онлайн без реєстрації. Його основна перевага абсолютна безкоштовність та інтеграція з Google Drive. Проте побудова UML-діаграм у Diagrams.net потребує більше ручної роботи: немає чітко структурованих UML-шаблонів, а вигляд елементів часто доводиться налаштовувати самостійно.

Lucidchart – це хмарна онлайн-платформа для створення блок-схем, технічних схем і UML-діаграм. Вона має інтуїтивно зрозумілий інтерфейс, готові шаблони для різних типів діаграм, а також можливість експорту зображень у форматах PNG, JPG або PDF. Lucidchart не потребує встановлення, працює прямо у браузері та дозволяє швидко створювати структуровані діаграми навіть початківцю. Для студентських цілей безкоштовної версії цілком достатньо.

Враховуючи поставлені цілі кваліфікаційної роботи магістра, було обрано саме Lucidchart як основний інструмент для створення діаграм. Він поєднує в собі простоту, функціональність і зручність. Важливо й те, що діаграми можна швидко стилізувати відповідно до вимог до оформлення, експортувати в потрібному розширенні та без проблем експортувати в пояснювальну записку.

Ці діаграми дозволяють описати логіку взаємодії користувачів із платформи, структуру основних компонентів та основні процеси, що відбуваються всередині системи.

Отже, вибір Lucidchart є обґрунтованим рішенням, що повністю відповідає цілям і масштабу роботи, забезпечуючи наочність, зручність і відповідність академічним вимогам. Для доцільного обґрунтування вибору інструменту було проведено дослідження функціоналу і створено таблицю 2.1 для наочної демонстрації результатів.

Таблиця 2.1 – Порівняння інструментів для побудови діаграм

Критерій	StarUML	Lucidchart	Diagrams.net (draw.io)
Тип доступу	Десктопна програма з ліцензією	Онлайн-платформа з безкоштовним планом	Онлайн-платформа, повністю безкоштовна
Необхідність встановлення	Так	Ні	Ні
Підтримка UML-діаграм	Повна, професійна	Повна, базові типи	Часткова, вручну
Інтерфейс користувача	Просунутий, але складніший	Інтуїтивно зрозумілий, сучасний	Простий, функціональний
Шаблони для UML	Так, вбудовані	Так, багато готових шаблонів	Обмежено, переважно загальні блоки
Експорт діаграм	PNG, SVG, PDF	PNG, JPG, PDF	PNG, SVG, PDF

Ефективне моделювання є ключовим етапом розроблення програмного забезпечення, оскільки саме на цьому етапі відбувається формалізація вимог, визначення взаємодії користувачів із системою та структурування її внутрішніх компонентів. Використання UML забезпечує можливість описати систему до початку реалізації, уникнути неоднозначного трактування функціоналу та підвищити керованість процесу розроблення. Візуальні моделі спрощують

комунікацію між усіма учасниками розробки та дозволяють своєчасно виявити логічні помилки.

У даній роботі в рамках UML-моделювання розроблено діаграму варіантів використання (Use Case Diagram), діаграму класів (Class Diagram) та діаграму активності (Activity Diagram), що дозволяє візуалізувати сценарії роботи користувачів з платформою, архітектуру основних елементів системи та етапи виконання ключових операцій. Use Case Diagram – це модель, що представляє функціональні можливості програмної системи з точки зору операцій, доступних різним категоріям користувачів.

Головною метою побудови цієї діаграми є визначення того, які дії може виконувати кожний тип користувача, та в якому вигляді система реагує на ці дії. Вона дає можливість побачити систему «ззовні», сформувати чітке розмежування прав доступу й визначити межі відповідальності кожного актора. Такий підхід дозволяє чітко визначити, які саме варіанти використання платформи доступні кожному типу користувачів, а також як ці дії пов'язані між собою.

У межах проекту було побудовано діаграму варіантів використання, яка ілюструє основні сценарії взаємодії користувачів із веб-платформою онлайн-курсів іспанської мови. У системі передбачено два ключові актори: Учень та Адміністратор. Кожен з них має доступ до окремого набору функціональних можливостей відповідно до своєї ролі у системі. Нижче наведено рисунок 2.1, що демонструє як користувачі взаємодіють із основними функціями платформи та які сценарії її використання є найпоширенішими.

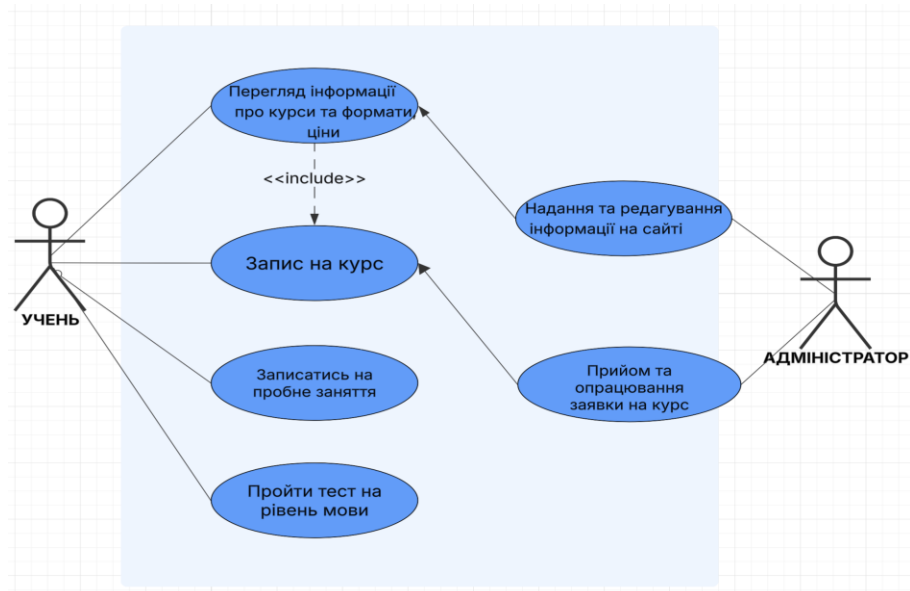


Рисунок 2.1 – Use case діаграма

*Джерело: розроблено автором*

Учень, як основний користувач, має змогу:

- перегляд доступних курсів (назва, рівень, формат навчання, вартість);
- проходження тесту на визначення рівня володіння мовою;
- запис на пробне заняття;
- подання заявки на повноцінний курс.

Адміністратор має інші функціональні можливості, зокрема:

- управління інформацією, розміщеною на платформі;
- перегляд та обробка заявок, наданих учнями;
- оновлення або редагування інформації про курси та їхні формати.

У діаграмі також використано зв'язки типу «include», які позначають, що певні дії є невід'ємною складовою інших. Наприклад, сценарій «Запис на курс» включає в себе дію «Записатись на пробне заняття», оскільки ці процеси логічно пов'язані між собою. Завдяки цій діаграмі стає можливим наочно продемонструвати загальну логіку функціонування системи, а також розмежування ролей між її користувачами.

Діаграма активностей відображає послідовність кроків, які виконує користувач під час взаємодії з веб-платформою. На відміну від діаграми варіантів використання, яка показує що може робити користувач, діаграма

активності демонструє як саме виконується сценарій – у якій послідовності відбуваються кроки, які етапи є обов'язковими, а які – альтернативними. Такий тип моделювання дозволяє представити логіку виконання дій у вигляді зрозумілої схеми, де кожен етап пов'язаний із попереднім і демонструє можливі шляхи руху в межах системи.

У створеній діаграмі активностей змодельовано типову поведінку користувача та учня від першого входу на сайт до початку проходження курсу. Діаграма відображає бізнес-логіку користувацького сценарію та містить як послідовні кроки, так і розгалуження залежно від дій користувача. Основні дії включають:

- 1) вхід на сайт;
- 2) ознайомлення з загальною інформацією про курси;
- 3) проходження тесту на визначення рівня знань (у разі потреби);
- 4) вибір бажаного курсу;
- 5) перегляд детальної інформації про курс;
- 6) запис на курс;
- 7) перехід до проходження курсу (виконання навчальних завдань на зовнішніх платформах – Miro або Zoom);
- 8) завершення взаємодії з сайтом (закриття сторінки).

Діаграма містить умовний перехід – на етапі проходження тесту, який не є обов'язковим. Решта дій виконується послідовно й відображає логічний сценарій типового користувача, який ознайомлюється з курсами, обирає програму й розпочинає навчання. Поданий рисунок 2.2 ілюструє послідовність дій, дозволених для користувачів платформи.

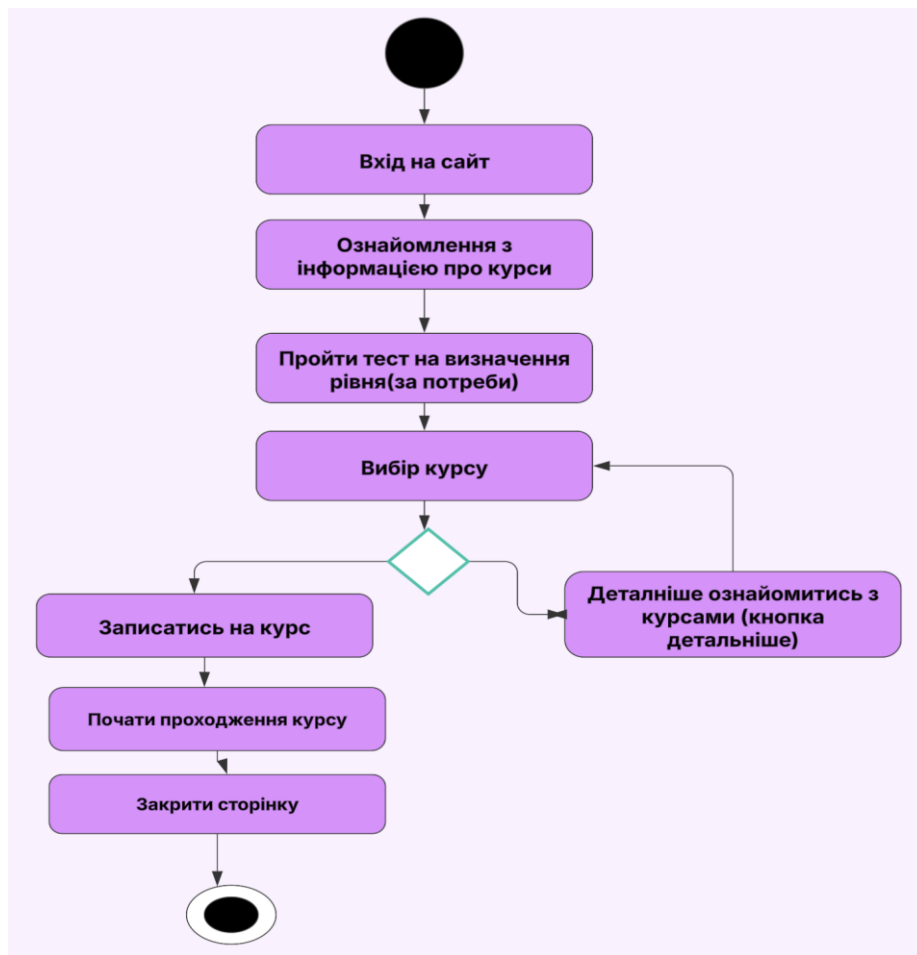


Рисунок 2.2 – Діаграма активності

*Джерело: розроблено автором*

Діаграма активності дала змогу формалізувати логіку взаємодії користувача з платформою та визначити послідовність етапів, які необхідно реалізувати в інтерфейсі системи. Вона також є підґрунтям для формування вимог до реалізації бізнес-процесів у програмній частині.

Діаграма класів (Class Diagram) – це один із базових типів UML-діаграм, який використовується для графічного зображення структури системи. Вона демонструє основні сутності (класи), їхні характеристики (атрибути), функціональність (методи), а також взаємозв'язки між цими елементами. Така діаграма дозволяє побачити внутрішню логіку системи ще до її реалізації, що особливо важливо на етапі проектування. На рисунку 2.3 зображено структуру системи, що визначає взаємозв'язки між класами та основні компоненти платформи.

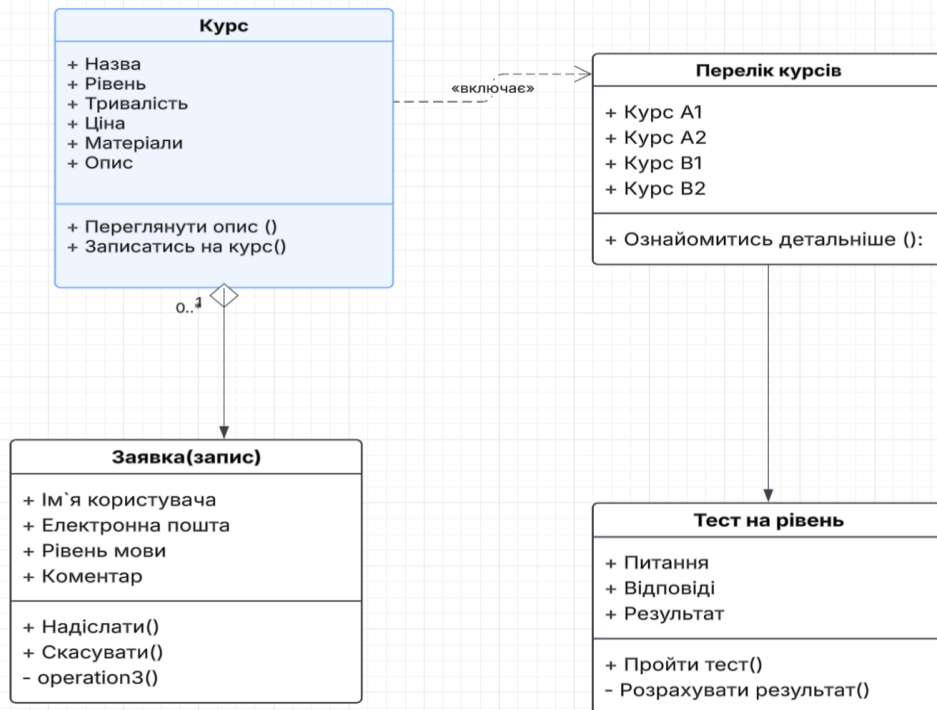


Рисунок 2.3 – Діаграма класів

*Джерело: розроблено автором*

На діаграмі класів, представлений у даній роботі, змодельовано структуру веб-платформи онлайн-курсів іспанської мови. Основними класами є: «Курс», «Заявка», «Тест на рівень» та «Перелік курсів». Клас «Курс» містить ключові атрибути, як-от назва, рівень, тривалість, ціна, матеріали й опис. З ним пов'язаний клас «Заявка», що зберігає інформацію про користувача та дозволяє надсилати або скасовувати запис. Також система передбачає проходження тесту для визначення рівня, що реалізується через окремий клас з відповідними атрибутами – питання, відповіді, результат. Перелік доступних курсів подано у вигляді окремого класу.

Діаграма відображає логічні зв'язки між класами відповідно до функціональних потреб сайту. Наприклад, один курс може мати кілька заявок (агрегація), а тест пов'язаний із процесом запису. Побудована модель стала основою для подальшої реалізації бази даних і структури взаємодії між компонентами системи.

## **2.3 Обґрунтування вибору шляхів, технологій та засобів вирішення поставленого завдання**

Для реалізації веб-платформи було обрано сучасний набір технологій, що забезпечує швидку, адаптивну та масштабовану реалізацію програмного забезпечення. Вибір інструментів здійснювався на основі вимог, сформованих у розділі 2. Система повинна бути адаптивною, підтримувати роботу з базою даних та забезпечувати можливість інтеграції нових модулів [27]. Архітектура системи побудована за клієнт-серверною моделлю та складається з трьох незалежних рівнів:

- frontend;
- backend API;
- database.

Архітектура платформи структурована у вигляді окремих модулів: інтерфейс – React, серверна частина – Node.js + Express, база даних – MongoDB Atlas та візуальні елементи – Tailwind CSS. Однією з основних ідей було зробити сайт, який буде інтуїтивно простим у користуванні та коректно виглядатиме на будь-якому пристрої. Усі використані технології є безкоштовними, популярними та мають багато навчальних матеріалів онлайн. Ось основний перелік:

- а) frontend;
- б) backend;
- в) база даних;
- г) інші інструменти.

Усі компоненти інтегруються між собою та дають можливість швидко реалізовувати функціонал, з яким легко працювати. У результаті ці інструменти дозволили зробити зручну, сучасну і надійну платформу, яку можна легко розширювати або змінювати в майбутньому.

Для реалізації клієнтської частини обрано бібліотеку React, оскільки вона забезпечує компонентний підхід, що спрощує розроблення та подальшу

підтримку системи. Компонентна модель дозволяє структурувати інтерфейс на незалежні модулі, які можна модифікувати без впливу на інші частини системи. Додатково варто зазначити, що хоча React використовує JSX замість класичного HTML, фактично JSX трансформується у стандартні HTML-елементи, які браузер обробляє нативно. Використання віртуального DOM забезпечує високу швидкість відображення елементів, що є важливим для веб-платформи з динамічними елементами та формами [28].

Фундаментом клієнтської частини платформи є три ключові веб-технології. HTML формує структуру сторінок і забезпечує коректне відображення елементів інтерфейсу в браузерах. CSS відповідає за зовнішнє оформлення: кольорову гаму, типографіку, розташування блоків та пристосування інтерфейсу до різних розмірів екранів. JavaScript реалізує динамічну поведінку системи: реакцію на дії користувача, перевірку введених значень у формах та передачу інформації між браузером і сервером. Ці технології створюють єдиний механізм, на основі якого працює бібліотека React для побудови компонентної архітектури додатка.

Для стилізації інтерфейсу застосовано Tailwind CSS. Вибір цієї технології обумовлений можливістю створення адаптивних інтерфейсів із мінімальними витратами часу на написання індивідуальних CSS-стилів. Tailwind дозволяє використовувати готові утилітарні класи, завдяки чому зменшується обсяг стилів і забезпечується візуальна цілісність дизайну [29]. Додатково використано React Router, що забезпечує маршрутизацію між сторінками без перезавантаження браузера, створюючи ефект односторінкового застосунку (SPA) та підвищуючи швидкість взаємодії користувача з системою.

Серверна частина реалізована з використанням Node.js у поєднанні з фреймворком Express. Обидві технології працюють на JavaScript, що забезпечує єдність мови програмування на клієнтському та серверному рівнях, зменшуючи складність проекту та скорочуючи витрати на підтримку [30]. Express використовується для створення REST-орієнтованого API, який обробляє HTTP-запити, отримані від веб-інтерфейсу, зокрема заявки користувачів на

навчання. Обмін даними між клієнтом і сервером здійснюється у форматі JSON, що спрощує інтеграцію модулів. Також серверна частина доповнена механізмами валідації даних та захисту від надмірної кількості запитів (rate limiting), що підвищує безпеку платформи та запобігає спробам спаму. Логіка оброблення заявок винесена в окремий контролер, а маршрути та middleware структуровано за принципами модульності, що спрощує підтримку та розширення серверного функціоналу. Такий підхід забезпечує стабільну роботу API та створює основу для подальшої інтеграції адміністративної панелі або інших сервісів.

Для роботи з базою даних обрано MongoDB Atlas – хмарну нереляційну базу даних, що дозволяє зберігати структуровані та напівструктуровані дані. Вибір обґрунтований підтримкою масштабованості, автоматичним резервним копіюванням та зручним механізмом інтеграції з Node.js через офіційний драйвер. Зберігання даних здійснюється у вигляді документів формату JSON, що забезпечує гнучкість структури даних [31].

Управління вихідним кодом, контроль версій та спільна робота з репозиторієм здійснюються засобами GitHub. Для тестування API-запитів використовувався Postman, що дозволило перевірити коректність передачі даних та роботу інтеграції з базою даних. Оскільки проєкт має академічний характер, веб-додаток розгорнуто локально на персональному комп'ютері для забезпечення зручності тестування та відлагодження. Такий підхід дозволяє оперативно вносити зміни, контролювати роботу серверної частини та бази даних, а також уникнути додаткових витрат на хмарну інфраструктуру під час етапу розроблення.

Застосований стек технологій забезпечує модульність, масштабованість та можливість подальшого розвитку систем, тобто додавання нових модулів не потребуватиме зміни вже реалізованих компонентів. Таким чином, обрані технології повністю відповідають вимогам до функціональності веб-платформи та забезпечують стабільність її роботи в реальних умовах експлуатації.

Для реалізації даного проєкту було обрано середовище розробки Visual Studio Code. Це зручний, безкоштовний редактор коду, який підтримує велику кількість мов програмування, має розширення для роботи з React, Tailwind, Node.js та дозволяє комфортно працювати з файлами, терміналом і відлагодженням прямо в одному вікні. Нижче наведено фото середовища розробки рисунок 2.4.

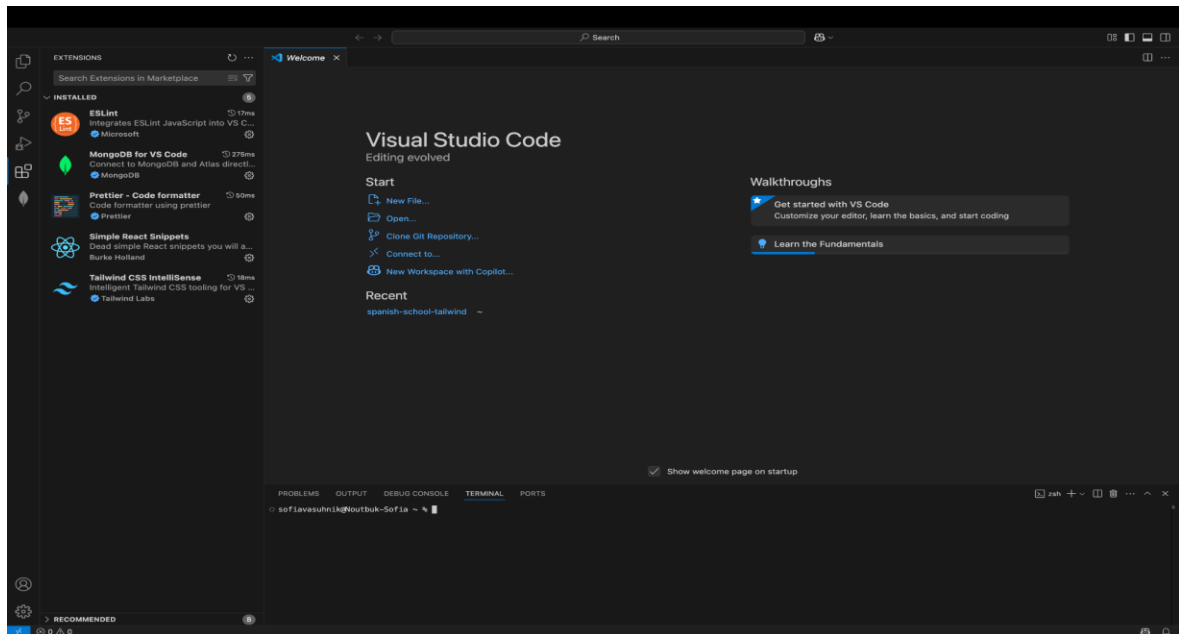


Рисунок 2.4 – Середовище розробки Visual Studio Code [32]

## 2.4 Практична реалізація веб платформи для надання курсів з іспанської

На початковому етапі було створено робоче середовище у Visual Studio Code та ініціалізовано фронтенд-проєкт за допомогою інструмента Vite. Через термінал встановлено необхідні технології – React, Tailwind CSS та React Router. Крім того, важливим етапом стало встановлення й налаштування допоміжних інструментів: Node.js, прт та модулів, що забезпечують роботу з формами, маршрутизацією та адаптивною версткою, які є стандартним набором для розроблення сучасних веб-інтерфейсів.

Загалом структура сайту реалізована у вигляді основних компонентних модулів. Основні структурні компоненти веб-платформи включають такі елементи, як: «Header», «Footer», головна сторінка, розділи «Курси», «Про курс», «Ціни», інтерактивний блок «Тест на визначення рівня» та модальне вікно для запису. Кожен з компонентів був створений як окремий файл і має власний код з логікою та стилями. Такий підхід забезпечив зручну організацію проєкту. Розроблений веб-додаток реалізовано у вигляді клієнт-серверної системи з чітким поділом логіки між фронтенд- і бекенд-частинами. Архітектура побудована на модульному принципі, що забезпечує гнучкість розширення та зручність супроводу.

В додатку Б подано структуру директорій веб-платформи. Така організація забезпечує модульність коду, зручне масштабування функціоналу та можливість швидкої адаптації системи для подальшої інтеграції з базою даних чи зовнішніми API. Усі компоненти організовано за функціональним призначенням, а структура директорій відображає логіку роботи клієнтської та серверної частин.

Для підвищення структурованості проєкт поділено на основні каталоги:

- `public/` – зберігає медіафайли (зображення, іконки, ілюстрації), які використовуються на сторінках платформи;
- `server/` – містить серверну логіку Node.js, конфігураційні файли та маршрути для обробки запитів;
- `src/` – основний каталог застосунку, де розміщено підкаталоги та файли стилізації;
- кореневі файли конфігурації (`package.json`, `vite.config.js`, `postcss.config.js`) визначають середовище збірки та залежності проєкту.

Відкривши сайт користувач потрапляє на головну сторінку (Home page) з безпосередньою кнопкою навігації на основні курси. Також розміщено Header з навігацією по сайту, логотипом та кнопками зв'язку. Для реалізації зручної навігації між сторінками було використано бібліотеку React Router. Рисунок 2.5 відображає головну сторінку сайту з навігацією.

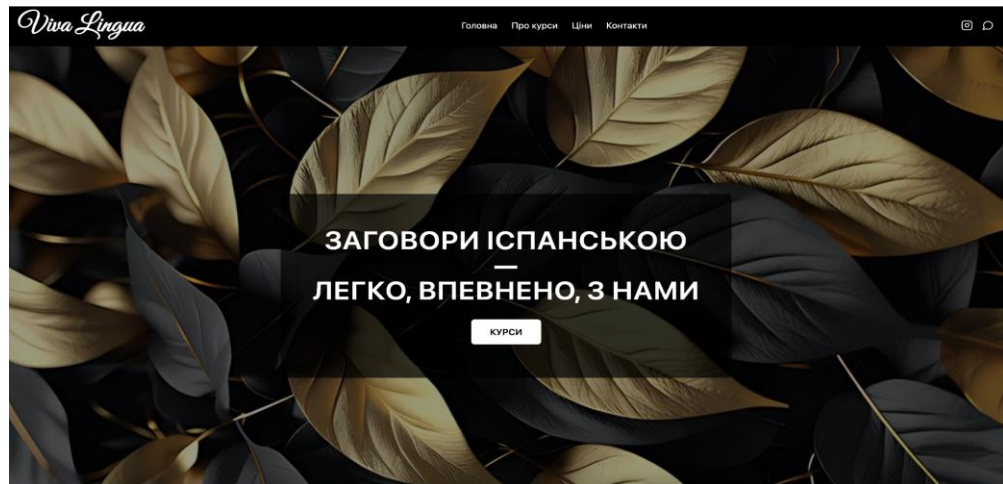


Рисунок 2.5 – Головна сторінка

*Джерело: розроблено автором*

Ключовий контент платформи – сторінки з головною інформацією «Курс А1», «Курс А2», «Курс В1», «Курс В2», які представлено окремим повноекранними блоками. Для їх реалізації використовувались React-компоненти, що забезпечують повторюваність структури та динамічне відображення даних. Стилізація та адаптивність інтерфейсу досягалися за допомогою Tailwind CSS, що дозволило швидко формувати єдину візуальну систему. Основними складовим структури цих модулів є чіткий, лаконічний опис та дві кнопки – «Детальніше» та «Записатись». Рисунок 2.6 демонструє основну логіку дизайну платформи.

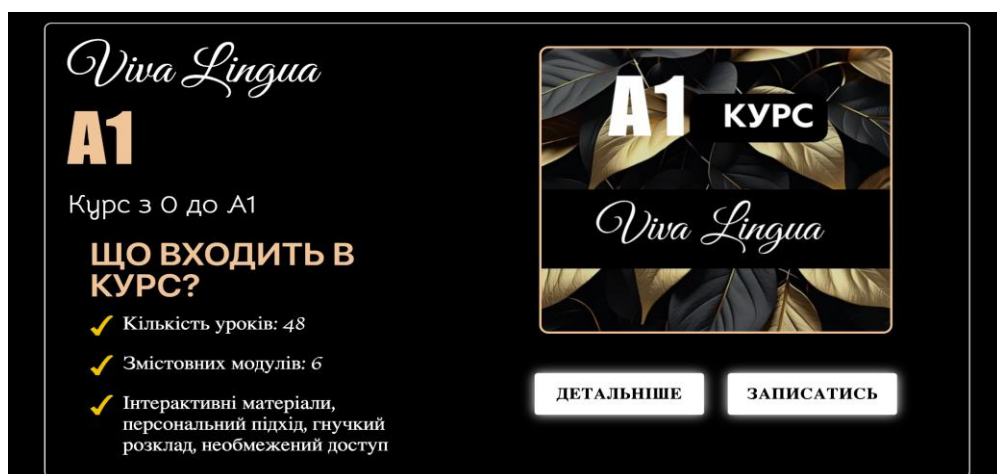


Рисунок 2.6 – Дизайн сайту

*Джерело: розроблено автором*

Для запису на курс на платформі було створено модальну форму, яка відкривається при натисканні кнопки «Записатись». У цьому компоненті реалізовано збирання введених даних, валідацію полів та відправлення інформації на серверну частину через HTTP-запит. Користувач заповнює свій рівень, номер телефону, нікнейм і за потреби залишає коментар. На рисунку 2.7 зображено реалізовану форму для запису на курс.

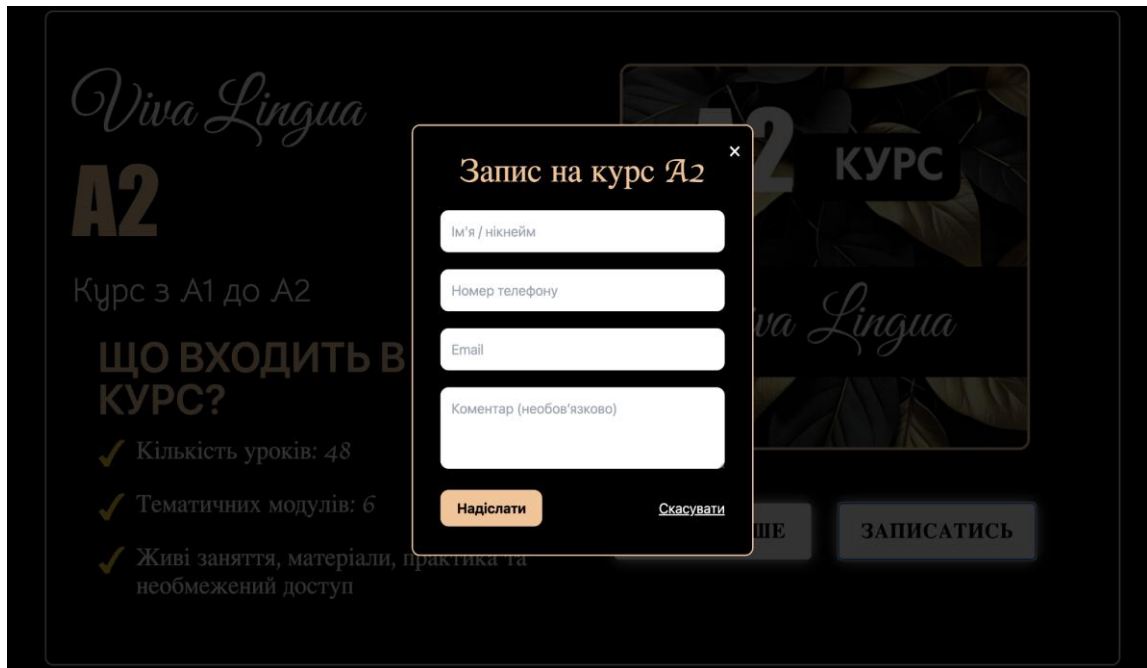


Рисунок 2.7 – Форма для запису на курс

*Джерело: розроблено автором*

Для демонстрації реалізації функціоналу подання заявки користувача на участь у курсі наведено фрагмент програмного коду, що відповідає за обробку форми запису та подано в додатку Б. Після натискання кнопки надсилання, дані автоматично передаються через Axios-запит на сервер, де обробляються Node.js + Express і зберігаються в базі даних MongoDB Atlas.

Складові компоненти сайту, такі як «Про курси», «Вартість», «Кому підходять наші курси», мають статичний інформативний характер і призначені для ознайомлення користувача з основною та загальною інформацією. Застосовані інструменти проектування забезпечують якісне масштабування цих

складових. Демонстрацію статичного характеру компонентів представлено на рисунку 2.8.

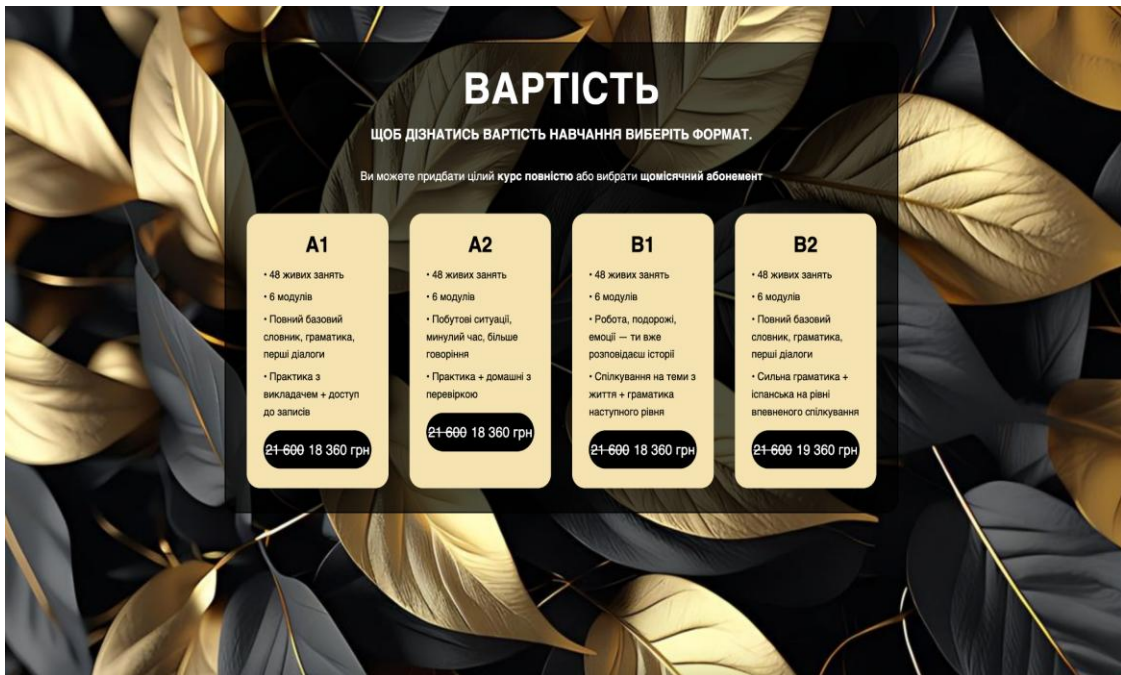


Рисунок 2.8 – Вартість курсів

*Джерело: розроблено автором*

В свою чергу, надання курсів забезпечується за допомогою інтеграції інтерактивного інструменту «Miro», що забезпечує найефективнішу гнучку та якісну подачу навчальних матеріал [33]. Інтеграція зовнішнього інструменту була обрана як найоптимальніший варіант, оскільки такий підхід забезпечує найвищу ефективність подання навчальних матеріалів: платформа підтримує інтерактивність, спільну роботу та миттєву взаємодію користувачів у реальному часі. Для їх реалізації використовувався фронтенд компонент React. Для маршрутизації і правильної організації переходів між сторінками та зовнішніми ресурсами використано React Router, що дозволяє відкривати інтерактивні інструменти без порушення логіки роботи SPA-додатку. Адаптивне відображення елементів інтерфейсу та стилізація кнопок інтеграції реалізовані за допомогою Tailwind CSS, який гарантує коректність відображення на різних пристроях та роздільностях екрана. На рисунках 2.9 та

2.10 зображено структуру організації курсів та приклад навчальних матеріалів, які відображає основні компоненти та логіку подання навчальних матеріалів.



Рисунок 2.9 – Структура організації курсів

*Джерело: розроблено автором*



Рисунок 2.10 – Приклад навчальних матеріалів

Джерело: розроблено автором

Додаткові опціональні частини, зокрема кнопки для проходження тесту на визначення рівня іспанської та запису на пробне заняття, були реалізовані за допомогою React і ведуть на відповідні Google Forms. Футер сайту створено з використанням інструментів стилізації, він містить контакти, соцмережі та посилання, і слугує для швидкої навігації та зв'язку з адміністрацією платформи.

## 2.5 Оцінка функціональних характеристик системи

Після завершення реалізації всіх ключових модулів веб-платформи було проведено оцінку функціональності компонентів з урахуванням цільового призначення системи, користувацького досвіду та технічної стабільності. Основна мета – переконатись, що всі складові сайту виконують покладені на них завдання, працюють коректно у звичних сценаріях користування та не створюють перешкод у взаємодії з потенційними учнями.

Головна мета освітньої платформи – надати користувачам доступну та чітку інформацію про курси іспанської мови, забезпечивши простий процес запису на навчання. Зручність у користуванні, зрозуміла структура і передбачувана навігація сприяють позитивному досвіду взаємодії з сайтом та заохочують потенційних учнів до реєстрації.

Усі ключові функції, передбачені на етапі планування платформи, були реалізовані повністю й працюють стабільно. Навігація, форма заявки, перегляд курсів, тестування рівня, перегляд цін та блоки контактів – усе це взаємодіє між собою, формуючи єдину, зручну систему. Саме тому при розробці інтерфейсу було особливо важливо досягти балансу між візуальною привабливістю та функціональністю.

У процесі створення веб-платформи важливу роль відіграв не лише функціонал, а й візуальна складова. Дизайн має створювати враження професійності, легкості та естетичної привабливості, що відповідає іміджу сучасної онлайн-платформи іспанської мови. Дизайн платформи розроблявся з чіткою метою – зробити сайт візуально приємним, зручним та простим для будь-якого відвідувача. Було важливо досягти балансу між професійним виглядом та легкістю у сприйнятті.

Основною стилістичною ідеєю стала комбінація чорного, білого та золотого кольорів – вона додає сайту елегантності та логічної структури. Усі елементи інтерфейсу витримані в єдиному стилі, нічого зайвого чи нав'язливого.

Дизайн платформи було розроблено з урахуванням зручності та інтуїтивної зрозумілості для користувача. Основні елементи інтерфейсу розташовані логічно та структуровано, що дозволяє відвідувачу швидко орієнтуватися у змісті сайту. Кнопки виклику до дії, розділи навігації та ключова інформація мають чітке візуальне виділення, що сприяє ефективній взаємодії користувача з платформою та забезпечує позитивний досвід взаємодії.

Візуальна складова підтримує загальну логіку взаємодії користувача з системою: кольорова палітра та мінімалістичність оформлення спрямовані на те, щоб користувач концентрувався на змісті платформи, а не на декоративних елементах. Усі сторінки витримані в єдиному стилі, що сприяє формуванню цілісного сприйняття системи.

Таким чином, проведена оцінка дала змогу підтвердити, що реалізовані функціональні компоненти відповідають заявленим вимогам, платформа працює стабільно, а її інтерфейс забезпечує позитивний користувацький досвід. Система готова до подальшого масштабування та тестування основних модулів.

## РОЗДІЛ 3

### ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ РЕЗУЛЬТАТИВНОСТІ РОЗРОБЛЕНОЇ ВЕБ-ПЛАТФОРМИ

#### 3.1 Методика проведення дослідження

Метою експериментального дослідження є перевірка результативності роботи розробленої веб-платформи для надання онлайн-курсів з іспанської мови, а також визначення, чи відповідає реалізований функціонал сформульованим вимогам.

Для забезпечення системності проведення експерименту було розроблено узагальнену методику дослідження, яка охоплює комплекс перевірок функціональних, нефункціональних та експлуатаційних характеристик веб-платформи [34]. Узагальнені відомості про використані підходи, їх мету, практичну реалізацію та критерії перевірки наведено в таблиці 3.1.

Таблиця 3.1 – Методика проведення експериментального дослідження результативності веб-платформи

Методика	Мета застосування	Реалізація	Критерій
Функціональне тестування	Перевірка роботи основних функцій та модулів	Виконано тести перегляду курсів, запису на курси і проходження тесту для перевірки рівня мови.	Коректність функціональної логіки системи
Нефункціональне тестування	Перевірка швидкодії, стабільності, зручності	Проведено заміри часу відповіді, перевірено адаптивність і стабільність	Продуктивність і якість користувацького досвіду
Вимірювання продуктивності	Визначення часу відгуку системи	Використано Chrome DevTools для аналізу часу запитів	Швидкодія серверних і клієнтських обробок
Перевірка достовірності даних	Переконалися у правильному збереженні інформації	Моніторинг колекцій у MongoDB Atlas	Надійність запису даних
Usability-тестування	Перевірка зручності інтерфейсу	Аналіз навігації та структури сторінок	Інтуїтивність користувацького досвіду

Для забезпечення достовірності та відтворюваності результатів дослідження було визначено та зафіксовано технічні й програмні умови проведення експерименту, які створювали стабільне середовище для перевірки роботи веб-платформи.

Експеримент проводився у локальному середовищі розробки на ноутбучі MacBook 15 Air M2 з операційною системою macOS Sequoia 15. Серверна частина платформи реалізована на Node.js з фреймворком Express, клієнтська частина – на React. Дані зберігались у хмарній базі MongoDB Atlas, а середовище розробки забезпечувалось за допомогою Visual Studio Code. Такий підхід дає можливість повністю відтворити реальну клієнт-серверну взаємодію незалежно від фактичного розміщення застосунку.

Для виконання дослідження було обрано такі інструменти:

- Chrome DevTools – для аналізу мережевих запитів та визначення часу їх виконання;
- Postman – для симуляції відправлення запитів на сервер та перевірки коректності відповіді API;
- MongoDB Atlas Monitoring – для відстеження появи нових записів у базі даних.

Об'єктом дослідження є ключова бізнес-функція платформи – запис користувача на курс через інтерактивну форму. Обрано саме цю операцію, оскільки вона поєднує всі рівні архітектури: інтерфейс користувача, API та базу даних.

Процедура експерименту виконувалася за таким алгоритмом:

- користувач відкриває веб-платформу у браузері;
- обирає курс та відкриває форму реєстрації;
- заповнює необхідні поля (ім'я, контактні дані, рівень підготовки);
- натискає кнопку відправлення заявки;
- форма ініціює HTTP-запит через Axios на backend-сервер;
- серверна частина обробляє запит та передає дані до MongoDB Atlas;
- у базі створюється новий запис, що підтверджує успішність операції;

– користувач отримує доступ до курсу.

Користувачу повертається відповідь про результат. Під час кожного циклу тестування фіксувалися такі параметри:

- час обробки запиту (від натискання кнопки до отримання відповіді);
- успішність передавання даних у базу (факт появи нового запису);
- стабільність роботи інтерфейсу (відсутність помилок або зависань).

Кількість повторень одного тестового сценарію – 10 ітерацій, що забезпечує достовірність отриманих результатів. Для уникнення впливу сторонніх факторів кожний тест виконувався при однакових умовах: однакове інтернет-з'єднання, той самий браузер та локальне оточення.

Методика дає змогу отримати кількісні та якісні показники роботи системи, які будуть проаналізовані у підрозділі 3.2. На основі зафіксованих даних здійснюється оцінювання результативності розробленої платформи та перевірка відповідності її функціональності висунутим вимогам.

### **3.2 Обробка та аналіз результатів дослідження**

Після завершення реалізації веб-платформи було проведено експериментальне дослідження її функціональної ефективності, зокрема перевірено швидкість обробки користувацьких запитів, успішність виконання дій користувачами та зручність взаємодії з інтерфейсом. Дослідження здійснювалось у локальному середовищі розробки на ноутбучі з використанням Visual Studio Code, Node.js + Express як серверної частини та бази даних MongoDB Atlas.

Основна мета – підтвердити коректність роботи функціональних модулів системи, перевірити швидкодію клієнт-серверної взаємодії, а також оцінити достовірність збереження даних у базі. Для збору даних використовувалися інструменти Chrome DevTools, Postman і MongoDB Atlas Monitoring, що забезпечили можливість об'єктивного вимірювання показників продуктивності, перевірки коректності запитів і контролю стану бази даних [35-36].

Функціональне тестування. Експериментальне моделювання функціональних можливостей платформи дозволило протестувати всі ключові функції веб-платформи. Перевірка проводилася за реальними сценаріями дій користувачів.

Однією з базових функціональних можливостей є виведення інформації про доступні курси. У ході експерименту перевірялося коректне відображення назв курсів, рівнів, тривалості навчання, формату занять та вартості. Після завантаження сторінки платформа надсилає запит до серверної частини, яка зверталась до бази даних. Під час тестування сторінки курсу рівня B1 у Chrome DevTools було зафіксовано коректне завантаження зображень та SVG матеріалів, кожен із яких отримав статус 200 ОК. Час відповіді сервера становив від 8 до 18 мс, що підтверджує швидке відображення мультимедійного контенту та оптимізовану роботу сторінки. На рисунку 3.1 продемонстровано результат тестування сторінки курсу B2 інструментом Chrome DevTools.

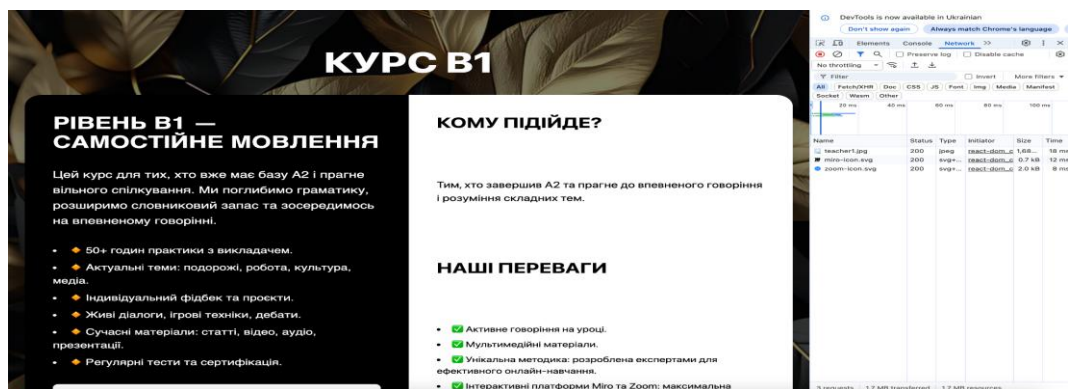


Рисунок 3.1 – Відображення курсів у браузері

*Джерело: розроблено автором*

Наступним тестованим елементом стала форма запису на пробне заняття. Користувач заповнював контактні дані, обирав курс та надсилав заявку, після чого система ініціювала POST-запит /apply. Під час експерименту було встановлено, що сервер обробляє запит і повертає відповідь за 109 мс, що є показником високої продуктивності. Валідація введених даних здійснюється на

стороні клієнта перед відправленням запити, що запобігає передачі некоректної інформації на сервер. Після успішного збереження заявки у базі даних користувач отримує підтвердження про прийняття запити, а адміністратор системи – відповідне сповіщення для подальшої обробки. На рисунку 3.2 зображено результати тестування.

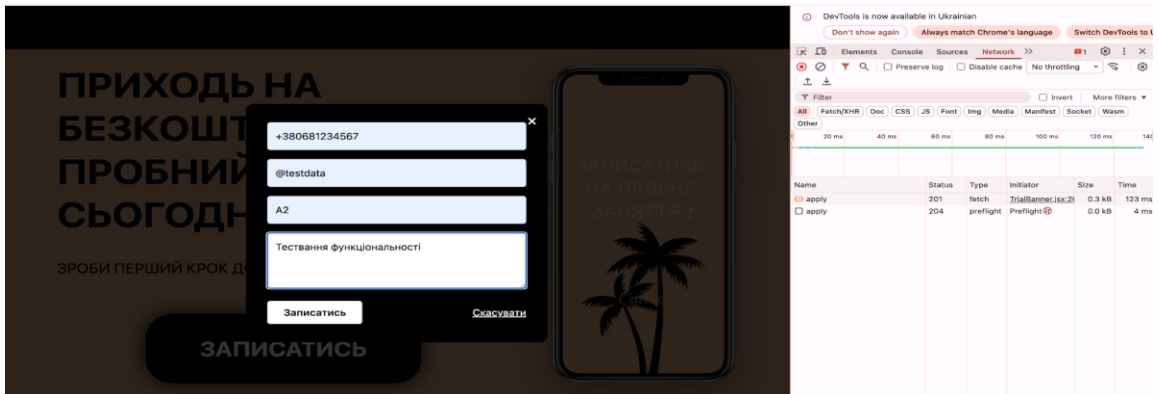


Рисунок 3.2 – Результат тестування запису на пробне заняття

*Джерело: розроблено автором*

Коректність обробки успішно перевірена за допомогою – відповіддю зі статусом 201 Created. Додатково у MongoDB Atlas зафіксовано появу нового документа в колекції applications, що доводить успішність запису та надійність збереження даних. Рисунок 3.3 відображає результат збереження заявок в базі даних.

```
_id: ObjectId('692defd47b1d126fbb18da34')
name: "СОЛОМІЯ"
email: "sofiastudent88@gmail.com"
course: "A2"
message: "Телефон: +380688387597. Коментар: Тест"
createdAt: 2025-12-01T19:43:16.713+00:00
updatedAt: 2025-12-01T19:43:16.713+00:00
__v: 0
```

```
_id: ObjectId('692defeb7b1d126fbb18da36')
name: "СОЛОМІЯ B2"
email: "sofiastudent88@gmail.com"
course: "B1"
message: "Телефон: +380688387597. Коментар: Тест"
createdAt: 2025-12-01T19:43:39.038+00:00
updatedAt: 2025-12-01T19:43:39.038+00:00
__v: 0
```

Рисунок 3.3 – Новий документ у колекції applications у MongoDB Atlas

*Джерело: розроблено автором*

В ході експериментально дослідження було також протестовано роботу навігаційних елементів інтерфейсу, зокрема хедера та футера, які містять кнопки для зв'язку з адміністрацією платформи. Під час натискання на відповідні іконки відбувалося коректне перенаправлення користувача до Telegram, Instagram, Email, TikTok. Жодних збоїв чи помилок маршрутизації зафіксовано не було, що підтверджує стабільну інтеграцію зі сторонніми сервісами та правильну роботу клієнтської частини інтерфейсу. На рисунку 3.4 відображено перевірку роботи хедера.



Рисунок. 3.4 – Перевірка коректної роботи хедера, футера та кнопок переходу до месенджерів

*Джерело: розроблено автором*

Одним із ключових функціональних модулів платформи є можливість подати заявку на запис на курс. Під час тестування користувач заповнював форму, вказуючи ім'я, контактні дані та бажаний курс, після чого здійснювався HTTP-запит на сервер. У Chrome DevTools фіксувався мережевий запит типу POST /apply, а середній час відповіді системи становив 254 мс. Через Postman було підтверджено правильність структури переданих даних та отримання статусу 201 Created, що підтверджує успішну обробку заявки на сервері. Далі у MongoDB Atlas фіксувалася поява нового запису в колекції applications, що гарантує збереження заявки у базі даних і виконання логіки функціоналу. Рисунок 3.5 ілюструє процес тестування функціоналу.

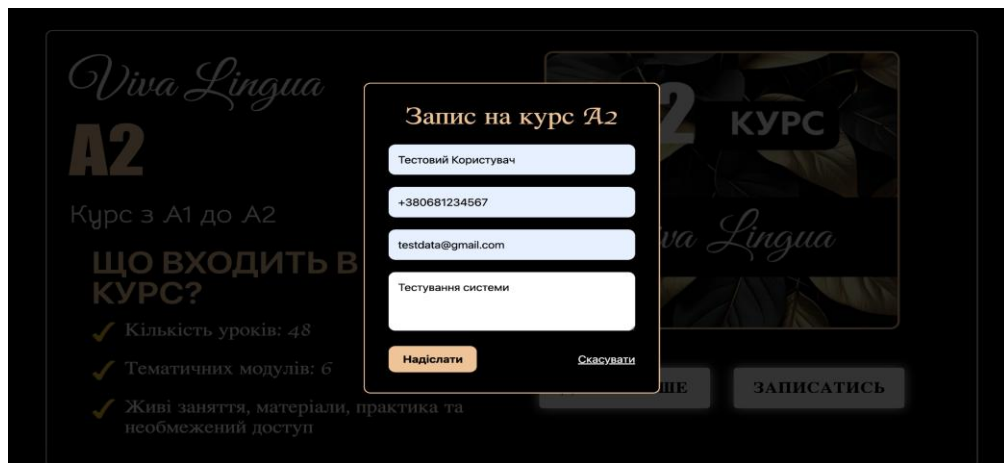


Рисунок 3.5 – Тестування функціоналу запису на курс

*Джерело: розроблено автором*

Для перевірки достовірності збереження даних було виконано моніторинг колекцій у хмарній базі даних MongoDB Atlas. На кожному етапі контролювалися цілісність структури об'єкта, правильність переданих полів та наявність часової позначки створення запису, що засвідчує надійність процесу збереження даних у системі. Рисунок 3.6 відображає базу даних MongoDB, в якій з'явився новий запис після подачі заявки.

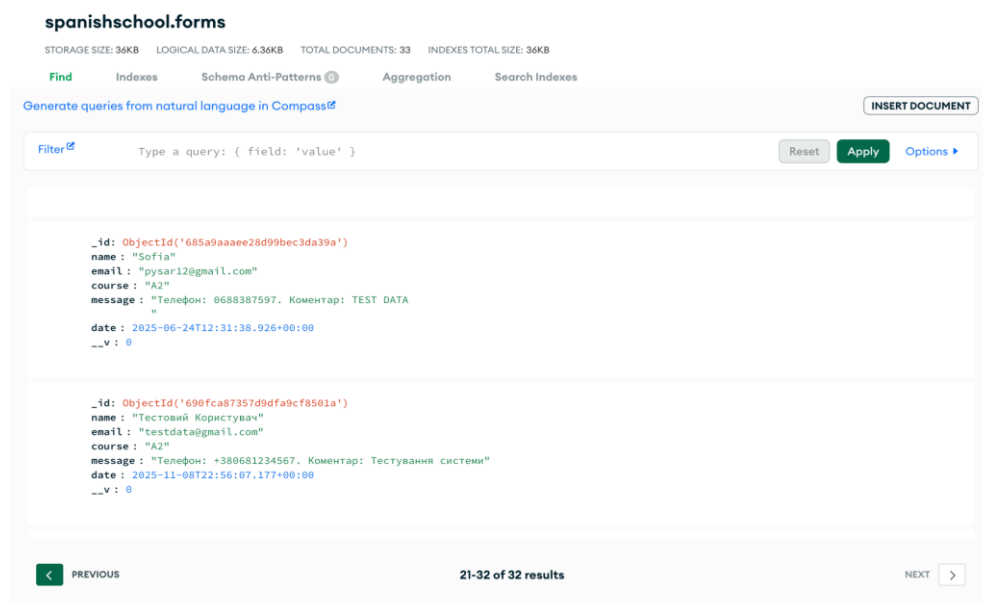


Рисунок 3.6 – Створення нового документа у MongoDB Atlas після подання заявки

*Джерело: розроблено автором*

У межах нефункціонального тестування було перевірено швидкодію, стабільність та зручність використання веб-платформи. Додатково перевірено адаптивність інтерфейсу. Поведінку елементів при зміні роздільної здатності підтвердила стабільне відображення контенту, а саме під час тестування система працювала без збоїв, сторінки завантажувалися коректно. За допомогою Chrome DevTools виконано заміри часу відповіді основних запитів, що дозволило оцінити продуктивність клієнт-серверної взаємодії. Додатково перевірено адаптивність інтерфейсу на різних розмірах екранів і поведінку елементів при зміні роздільної здатності, що підтвердило стабільне відображення контенту. Під час тестування система працювала без збоїв, сторінки завантажувалися швидко, що демонструє рисунок 3.7.

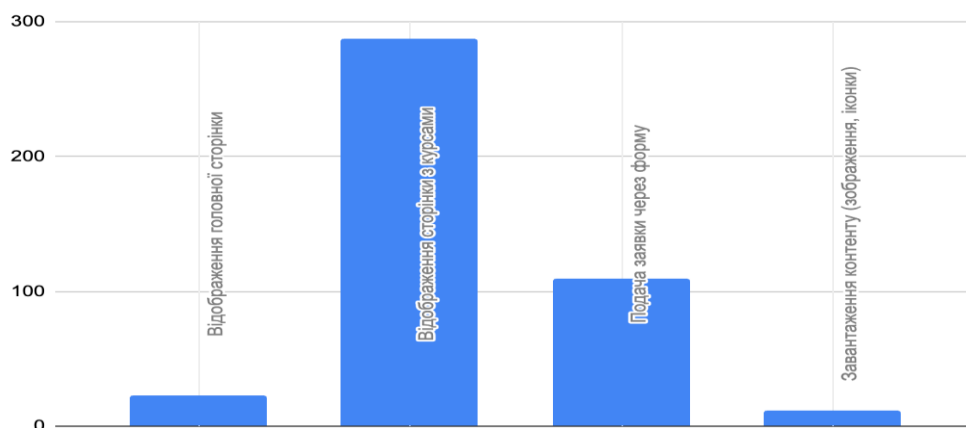


Рисунок 3.7 – Вимірювання часу виконання HTTP-запитів у Chrome DevTools (стовпчаста діаграма продуктивності системи)

*Джерело: розроблено автором*

Для оцінювання продуктивності веб-платформи було проведено вимірювання часу відгуку системи на ключові клієнтські та серверні запити. За допомогою інструмента Chrome DevTools виконувались заміри часу обробки запитів GET і POST, для визначення швидкодії та стабільності розробленої клієнт-серверної взаємодії. Отримані дані показали, що система обробляє запити швидко. Насамперед, завантаження головної сторінки відбувається за 23 мс, сторінки з курсами – за 287 мс, а подання заявки через форму показало

результат – 109 мс. Такі результати підтверджують високий рівень продуктивності та швидкості. На рисунку 3.8 наведено результати вимірювання часу виконання HTTP-запитів, отримані з інструменту Chrome DevTools.

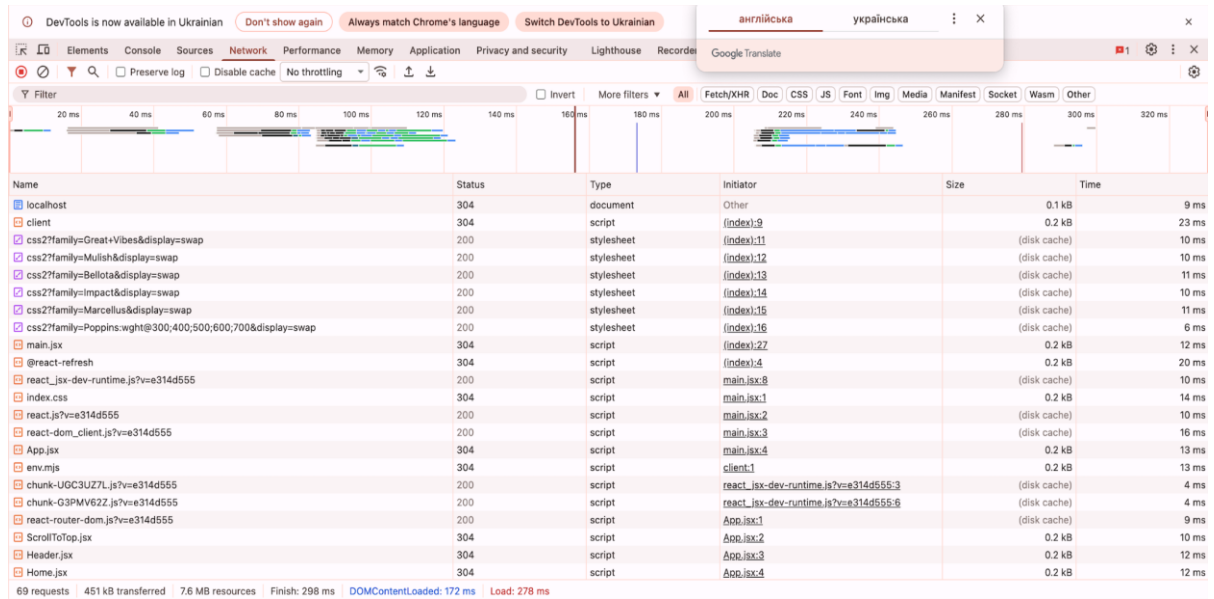


Рисунок 3.8 – Результати вимірювання часу виконання HTTP-запитів

Джерело: розроблено автором

Для об'єктивної оцінки ефективності розробленої платформи було проведено порівняльне тестування з чотирма провідними аналогами: Evermore, Espanita, Memrise та Alfa Grant. Вимірювання здійснювалися за ідентичною методикою в однакових технічних умовах, результати зведено в таблицю 3.2.

Таблиця 3.2 – Порівняльний аналіз продуктивності з існуючими аналогами

Назва платформи	Час завантаження головної (мс)	Час обробки форми (мс)	Час відображення курсу(мс)
Viva Lingua	23	109	287
Evermore	45	180	320
Espanita	38	165	295
Memrise	92	240	410
Alfa Grant	35	145	305
Google Web Vitals	<100	<100	<2500

Проведене порівняльне дослідження продуктивності розробленої веб-платформи Viva Lingua з провідними аналогами на ринку онлайн-освіти засвідчило її високі технічні характеристики. Час завантаження головної сторінки становить 23 мс, що на 49 % швидше за найближчого конкурента. Розроблена архітектура на базі React з оптимізованою серверною частиною Node.js забезпечила середній час відгуку системи на 35 % кращий середнього показника конкурентів (215 мс) та повністю відповідає рекомендаціям провідних технологічних компаній щодо якості користувацького досвіду [37]. Отримані результати підтверджують технологічну перевагу обраного стеку технологій і свідчать про високий рівень оптимізації програмного коду розробленої системи.

Для перевірки достовірності збереження даних було виконано моніторинг колекцій у хмарній базі даних MongoDB Atlas. Після відправлення форми користувача система ініціювала POST-запит, а у відповідній колекції бази з'явився новий документ із переданими даними. Це підтверджувало, що інформація з форми записується коректно, без втрат і дублювання. На кожному етапі контролювалися цілісність структури об'єкта, правильність переданих полів та наявність часової позначки створення запису, що засвідчує надійність процесу збереження даних у системі.

У межах usability-тестування доцільно оцінити зручність використання веб-платформи з позиції кінцевого користувача [39]. Аналізувались навігація, логіка переходів між сторінками, доступність ключових елементів інтерфейсу та інтуїтивність розміщення основних функцій. Під час тестування встановлено, що структура сторінок є зрозумілою: доступ до інформації про курси, форми подачі заявки та кнопок зв'язку не потребує додаткових пояснень. Елементи інтерфейсу реагують без затримок, форми заповнюються без перезавантаження сторінки, а адаптивний дизайн забезпечує коректне відображення на різних розмірах екранів. Результати підтверджують, що платформа є інтуїтивно зрозумілою та зручною у використанні, що підвищує якість користувацького досвіду. Нижче подано рисунок 3.9, що зображує

інтерфейс головної сторінки веб-платформи з відображенням структури, навігації та основних елементів дизайну.

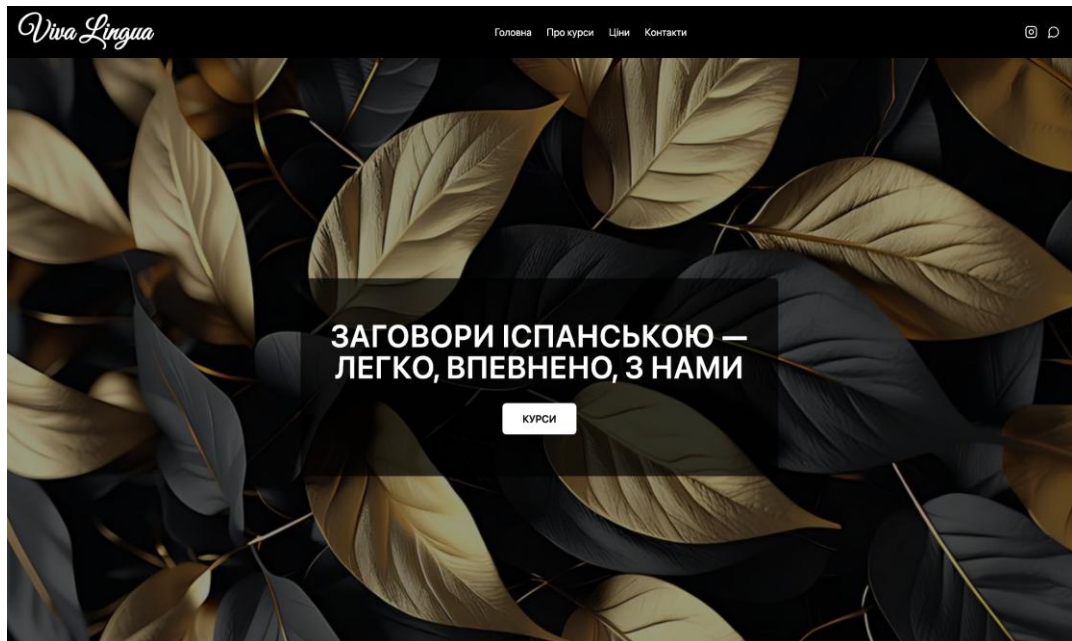


Рисунок 3.9 – Інтерфейс головної сторінки веб-платформи з відображенням структури, навігації та основних елементів дизайну

*Джерело: розроблено автором*

Інтеграція навчального курсу на платформу відбувається без затримок і технічних помилок, що свідчить про коректну взаємодію всіх клієнтських та серверних компонентів. Завантаження матеріалів, підключення зовнішніх інструментів і відтворення контенту здійснюються стабільно, без втрати даних чи погіршення якості роботи інтерфейсу. На рисунку 3.10 подано результати перевірки коректності інтеграції

Отримані результати експериментального дослідження підтвердили ефективність та надійність роботи веб-платформи для надання онлайн-курсів з іспанської мови. Система стабільно виконує всі основні функції, забезпечує швидку обробку запитів і коректне збереження даних у базі. Інтерфейс демонструє швидке реагування на дії користувача, а архітектура клієнт-серверної взаємодії виявилася оптимальною для стабільної роботи навіть за активного обміну даними між компонентами системи.

The image shows a digital learning interface for Spanish. The left side contains grammar exercises:

- 1. Qué actividades puedes hacer en estos lugares?** (What activities can you do in these places?) with images of a park, a library, a museum, a cafe, a gym, and a cinema.
- 2. Relaciona las actividades con las fotos.** (Relate the activities with the photos.) with a grid of activity icons.
- 3. ¿Con qué frecuencia haces estas actividades?** (How often do you do these activities?) with a frequency scale from 'nunca' to 'siempre'.
- 4. Escucha y responde.** (Listen and answer.) with a yellow box for the answer.
- 5. Lee y subraya cómo hablas de los gustos.** (Read and underline how you talk about tastes.) with a text passage about hobbies and preferences.
- 6. Completa las frases con un pronombre personal.** (Complete the sentences with a personal pronoun.) with a list of pronouns and a text passage.

The right side features a social media simulation:

- Abre las cartas y forma frases.** (Open the cards and form sentences.)
- Haz la frase.** (Make the sentence.) with a text box.
- Me gusta hablar de los gustos favoritos. Lee y relaciona.** (I like talking about favorite tastes. Read and relate.) with a matching exercise between names and preferences.
- Me encanta / Me gusta mucho / Me gusta bastante / Me gusta / No me gusta mucho / No me gusta nada** (I love / I like a lot / I like a lot / I like / I don't like a lot / I don't like at all) with corresponding Russian translations.
- Encuentra en el texto a las personas con las que coincidías en los gustos.** (Find in the text the people you coincided with in tastes.) with a text passage and a matching exercise.
- Me encanta el cine. / A mi también. / Pues a mí no. / No me gusta montar en bicicleta. / A mí tampoco. / Pues a mí sí.** (I love cinema. / Me too. / Not me, I don't like cycling. / Neither do I. / Yes, I do.) with a matching exercise.
- Buscamos según los gustos.** (We search according to tastes.) with a list of hobbies and a corresponding text passage.

Рисунок 3.10 – Результати перевірки коректності інтеграції

Джерело: розроблено автором

Проведені вимірювання продуктивності, аналіз мережевих запитів і перевірка відображення інформації показали, що платформа відповідає сучасним вимогам до швидкодії, зручності та точності передачі даних. Отже, розроблена система є готовим до впровадження інструментом, який може ефективно використовуватись у процесі організації та проведення онлайн-навчання.

## ВИСНОВКИ

У результаті виконання кваліфікаційної роботи магістра проведено комплексне дослідження та розроблення веб-платформи для організації онлайн-курсів з іспанської мови. Виконано всі поставлені завдання, творено функціональну, технологічно обґрунтовану та практично придатну систему, яка забезпечує ефективну взаємодію користувача з навчальним контентом і засобами адміністрування.

У ході дослідження розглянуто сучасні аспекти онлайн-навчання проаналізовано наявні технічні рішення у сфері мовної освіти. Це дало змогу визначити особливості організації онлайн-курсів, виявити ключові тенденції цифрової освіти та оцінити їх вплив на процес створення веб-платформи. Проведений аналіз дозволив сформувані обґрунтовані вимоги до структури та функціональності майбутньої системи.

Визначено, структуровано та формалізовано вимоги до веб-платформи відповідно до її цільового призначення. Це забезпечило цілісність, логічність та послідовність розробки, а також дало можливість створити основу для проєктування архітектури, оцінювання функціональних можливостей та подальшого тестування системи. Вимоги охоплювали структуру, функціональні модулі, критерії продуктивності, зручності та надійності.

Обґрунтовано вибір технологічного стеку React – Node.js – Express – MongoDB Atlas – Tailwind CSS. Здійснений аналіз технологій підтвердив доцільність використання обраних інструментів: вони забезпечують масштабованість, продуктивність, модульність і гнучкість реалізації. Вибір архітектурного підходу «клієнт-сервер» дозволив оптимізувати обмін даними та спростити подальше розширення функціональності системи.

Розроблено UML-моделі, що відображають логіку роботи платформи та взаємодію її користувачів. Побудовані діаграми варіантів використання, активності та класів дали змогу формально описати бізнес-процеси системи, визначити структуру даних і взаємозв'язки між компонентами. Це сприяло

якісній організації етапу реалізації та усуненню можливих логічних невідповідностей.

Реалізовано веб-платформу відповідно до визначених функціональних та нефункціональних вимог. У межах практичної частини створено модулі презентації курсів, визначення рівня, запису на заняття, обробки заявок та комунікації з адміністратором. Реалізовані рішення демонструють узгодженість між аналітичною моделлю та програмною реалізацією, що підтверджує коректність проведених проєктних рішень.

Проведено тестування функціональних компонентів, що дозволило оцінити їх відповідність вимогам. Тестування охоплювало перевірку стабільності роботи інтерфейсу, коректності обміну даними між клієнтською та серверною частинами, а також перевірку механізмів збереження інформації. Отримані показники часу відгуку та продуктивності підтвердили відповідність системи сучасним стандартам веб-розробки.

Виконано експериментальне дослідження результативності веб-платформи та проаналізовано отримані дані.

Систему оцінено в умовах реального сценарію використання. Результати експерименту підтвердили працездатність розроблених модулів, стабільність функціональної логіки та коректність роботи механізмів запису на курс. Це доводить придатність платформи для практичного застосування у мовній школі.

Сформовано рекомендації щодо подальшого розвитку та вдосконалення системи.

До наукових і прикладних результатів роботи належить проєктування архітектури освітньої системи, побудова UML-моделей функціональних процесів, розроблення клієнтсько-серверної структури, оптимізованої під сучасні веб-стандарті. Практичне впровадження результатів дає можливість створення реального освітнього ресурсу для організації мовного навчання онлайн.

Результати роботи пройшли апробацію через публікацію статті у «Студентському науковому віснику Луцького національного технічного університету», що засвідчує їх відповідність вимогам наукової спільноти. Отримана апробація підтверджує достовірність висновків і практичну значущість проведеного дослідження.

Перспективними напрямками визначено впровадження інструментів платіжної системи, кабінета студента розширення переліку курсів та застосування штучного інтелекту для автоматизації. Застосування таких рішень сприятиме підвищенню ефективності цифрової освіти та розвитку освітніх технологій.

Загалом результати роботи підтверджують досягнення поставленої мети та виконання всіх завдань дослідження. Створена веб-платформа має наукову й практичну цінність, може бути основою для подальших досліджень у сфері дистанційного навчання та здатна до використання у реальному освітньому середовищі.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Калугін Р. Ю. Онлайн освіта: перспективи та проблеми в умовах військових конфліктів. Педагогічний дискурс. 2022. № 49(2). С. 245-252. DOI: <https://doi.org/10.32843/2663-6085/2022/49.2.37>.
2. Online Learning Statistics 2025 – Growth Key Trends. Coursmos. URL: <https://www.coursmos.com/online-learning-statistics/> (дата звернення: 27.11.2025).
3. E-learning: global market size by segment. Statista. URL: <https://www.statista.com/statistics/1130331/e-learning-market-size-segment-worldwide/> (дата звернення: 08.09.2025).
4. Coursera. Online Courses. URL: <https://www.coursera.org> (дата звернення: 12.09.2025).
5. Udey. Online Learning Platform. URL: <https://www.udemy.com> (дата звернення: 15.09.2025).
6. edX. Online Courses. URL: <https://www.edx.org> (дата звернення: 15.09.2025).
7. Duolingo. Language Learning Platform. URL: <https://www.duolingo.com> (дата звернення: 15.09.2025).
8. Онлайн освіта: перспективи та проблеми в умовах військових конфліктів. URL: [http://www.innovpedagogy.od.ua/archives/2022/49/part\\_2/37.pdf](http://www.innovpedagogy.od.ua/archives/2022/49/part_2/37.pdf) (дата звернення: 28.02.2025).
9. Міністерство цифрової трансформації України. 99 % українців вважають, що потрібно вивчати іноземні мови. URL: <https://thedigital.gov.ua/news/education/99-ukraintsiv-uvazhayut-shcho-potribno-vivchati-inozemni-movi-doslidzhennya-mintsifri-usaid-ta-kmis> (дата звернення: 18.09.2025).
10. Sky Lingua. Поширеність мов: 11 найпопулярніших мов світу. URL: <https://sky-lingua.com/ua/12-naypooshirenishih-mov-u-sviti/> (дата звернення: 10.09.2025).

11. Dabbish L., Stuart C. Designing for Online Language Learning: Principles and Practice. Journal of Educational Technology. 2021. Vol. 15, No. 3. P. 45-62.
12. García P., López M. Enseñanza práctica del español como lengua extranjera: métodos y recursos digitales. Madrid: Editorial Edinumen, 2020. 284 p.
13. Optima School. Різниця між онлайн та дистанційним навчанням. URL: <https://optima.school/blog/riznica-miz-onlajn-ta-distancijnim-navcannam> (дата звернення: 28.02.2025).
14. Evermore Spanish. Official Website. URL: <https://evermorespanish.com> (дата звернення: 19.09.2025).
15. Espanita – Spanish Online Courses. URL: <https://espanita.com> (дата звернення: 19.09.2025).
16. Memrise. Language Learning Platform. URL: <https://www.memrise.com> (дата звернення: 21.09.2025).
17. Alfa Grant. Online Learning Platform. URL: <https://alfagrant.com> (дата звернення: 22.09.2025).
18. Калінін А. Ю. Основи UI/UX дизайну: навч. посіб. Київ: КНЕУ. 2021. 184 с.
19. Баран С. В. Основи web-програмування: навч. посіб. Кривий Ріг: Державний ун-т економіки і технологій. 2023. 316 с.
20. Захарченко Н. Веб-програмування. Київ: Центр учбової літератури, 2020. 256 с.
21. Васильєв О. Програмування мовою PHP: навч. посіб. Київ: Видавництво Ліра-К. 2022. 368 с.
22. Костюченко А. О. Основи програмування мовою Python : навч. посіб. Чернігів: ФОП Баликіна С. М. 2020. 180 с.
23. Flanagan D. JavaScript: The Definitive Guide. 7th ed. O'Reilly Media. 2020. 230 p.
24. Соммервілл І. Інженерія програмного забезпечення. 10-те вид. Київ: Діалектика. 2021. 864 с.

25. UML-діаграми: навіщо потрібні та як їх створити. URL: <https://foxminded.ua/uml-diagramy/> (дата звернення: 28.11.2025).
26. Кайзер Г. UML-моделювання для студентів: посіб. Львів: ЛНУ. 2020. 156 с.
27. Мельник О. Г. Архітектура сучасних веб-додатків: навч. посіб. Київ: КНУ. 2022. 175 с.
28. React Documentation. React 18. URL: <https://react.dev/learn> (дата звернення: 10.11.2025).
29. Tailwind CSS Documentation. URL: <https://tailwindcss.com/docs> (дата звернення: 10.11.2025).
30. Node.js Documentation. URL: <https://nodejs.org/en/docs> (дата звернення: 10.11.2025).
31. MongoDB Documentation. MongoDB Manual Version 6.0 . URL: <https://www.mongodb.com/docs/manual/> (дата звернення: 10.11.2025).
32. Microsoft. Visual Studio Code: source code repository. URL: <https://github.com/microsoft/vscode>
33. Європейська мовна політика і цифрова освіта: монографія/ред. Т. В. Шевченко. Київ: Наук. Думка. 2022. 186 с.
34. Hunko I. Software testing in 2023: new trends and challenges. Herald of Kyiv Institute of Business and Technology. 2023. Vol. 49, No. 1-2. P. 25-36. DOI: <https://doi.org/10.37203/kibit.2023.49.04>.
35. Chrome DevTools Documentation. URL: <https://developer.chrome.com/docs/devtools/> (дата звернення: 10.11.2025).
36. Postman Documentation. URL: <https://www.postman.com/> (дата звернення: 10.11.2025).
37. Web Vitals: Essential metrics for a healthy site. Google Developers. URL: <https://blog.chromium.org/2020/05/introducing-web-vitals-essential-metrics.html>

## **ДОДАТКИ**

## ДОДАТОК А

В додатку А представлено апробацію наукових досліджень

УДК 004.738.5:378.147

Васюхник С.О., ст. гр. КНм-21

Науковий керівник к.т.н., доц Лук'янчук Ю.А.

Луцький національний технічний університет

### **ДОСЛІДЖЕННЯ ТА РОЗРОБКА ВЕБ-ПЛАТФОРМИ ДЛЯ ЕФЕКТИВНОГО НАДАННЯ ОНЛАЙН-КУРСІВ З ІСПАНСЬКОЇ МОВИ**

У статті розглянуто процес дослідження та розробки веб-платформи для онлайн-курсів іспанської мови з використанням методології Rational Unified Process. Проаналізовано сучасний стан онлайн-освіти в Україні та світі, особливості викладання іноземних мов у цифровому середовищі. Визначено функціональні та нефункціональні вимоги до платформи, формалізовано ролі користувачів. Обґрунтовано вибір технологічного стеку React.js, Node.js та MongoDB для реалізації системи. Представлено архітектуру платформи та основні компоненти користувацького інтерфейсу.

**Ключові слова:** веб-платформа, онлайн-освіта, іспанська мова, React.js, Node.js, RUP, адаптивний дизайн.

Vasiukhnyk S.

### **RESEARCH AND DEVELOPMENT OF A WEB PLATFORM FOR EFFECTIVE PROVISION OF ONLINE SPANISH LANGUAGE COURSES**

The article examines the process of research and development of a web platform for an online Spanish language school using the Rational Unified Process methodology. The current state of online education in Ukraine and the world, as well as the features of teaching foreign languages in the digital environment, are analyzed. Functional and non-functional requirements for the platform are defined, and user roles are formalized. The choice of the React.js, Node.js, and MongoDB technology stack for system implementation is substantiated. The platform architecture and main user interface components are presented.

**Keywords:** web platform, online education, Spanish language, React.js, Node.js, RUP, UML modeling, adaptive design.

**Постановка проблеми.** Сучасний світ характеризується стрімкою цифровізацією усіх сфер життя, зокрема й освіти. Онлайн-навчання перетворилося з альтернативного на повноцінний формат освітнього процесу, особливо після пандемії COVID-19 та в умовах воєнного стану в Україні. Водночас зростає попит на вивчення іноземних мов, зокрема іспанської, якою володіють понад шестисот мільйонів людей у світі. Проте існуючі платформи для мовного навчання часто не враховують специфіки українського ринку, потребують складної авторизації або не забезпечують належного рівня інтерактивності. Відповідно особливої актуальності набуває розробка спеціалізованої веб-платформи для онлайн-курсів іспанської мови, яка поєднувала б інформативність, зручність використання та ефективні засоби комунікації між викладачами, адміністрацією та потенційними учнями.

**Аналіз останніх досліджень і публікацій.** Питання проектування й розробки освітніх веб-ресурсів досліджував Грінченко О. А., який розглядав методології проектування інформаційних систем. Савінов О. Ю. висвітлював

особливості створення сучасних веб застосунків, та Бочкарьова Т. І. аналізувала методи моделювання інформаційних систем. Теоретичні аспекти інженерії програмного забезпечення представлені у працях Соммервілла І., а питання UML-моделювання детально розкриті Кайзером Г. Технологічні аспекти розробки на React.js та Node.js висвітлені Брауном М. та Браунлі Дж. відповідно. Проте комплексний підхід до створення спеціалізованих платформ для мовних шкіл з урахуванням методології RUP та сучасних веб-технологій залишається недостатньо дослідженим.

**Цілі статті.** Метою дослідження є обґрунтування підходів до проектування та практичної реалізації веб-платформи для онлайн-курсів іспанської мови з використанням методології Rational Unified Process та сучасного технологічного стеку React.js, Node.js, MongoDB. Завдання дослідження включають аналіз стану онлайн-освіти та ринку мовних курсів, формалізацію ролей користувачів та визначення функціональних вимог до системи, обґрунтування вибору технологій та архітектурних рішень, розробку компонентної структури інтерфейсу та клієнт-серверної взаємодії, проведення тестування продуктивності та функціональності платформи.

**Виклад основного матеріалу дослідження.** Світова освітня система перебуває в процесі трансформації під впливом цифрових технологій. Онлайн формат став не лише трендом, а й невід'ємною складовою освітнього процесу у сфері іноземних мов. Згідно з дослідженнями, глобальний ринок онлайн-освіти у 2021 році оцінювався у 250 мільярдів доларів, а до 2028 року прогнозується його зростання до 1 трильйона доларів. Така ситуація зумовила збільшення попиту на дистанційні курси, спеціалізовані веб-платформи та інтерактивні навчальні сервіси. Зважаючи на освітню специфікацію даної розробки слід зазначити, що курси є невід'ємною складовою сучасної онлайн-освіти. Провідні університети світу активно інтегрують онлайн-навчання у свої програми, що свідчить про високий рівень довіри до цього формату. [8]

В Україні цифрова трансформація освіти стала особливо актуальною в період пандемії COVID-19 та повномасштабного вторгнення. Зросла кількість як державних ініціатив, так і приватних проєктів, орієнтованих на цифрове навчання. Сегмент онлайн-мовних платформ продовжує зростати через високий попит серед українців, які вимушено залишили країну або готуються до переїзду.

Особливої уваги заслуговує динаміка сегмента онлайн-курсів з вивчення іноземних мов. Даний сектор продовжує стабільно зростати завдяки впровадженню сучасних цифрових інструментів, адаптивних навчальних систем та інтерактивних платформ. За даними аналітичних агентств, українці активно вивчають англійську, німецьку, польську та іспанську мови. Попит на вивчення іноземних мов зріс на 45 відсотків порівняно з 2019 роком, що пов'язано з міграційними процесами, необхідністю працевлаштування за кордоном та прагненням до міжнародної освіти.

- Фаза початку передбачає визначення основної мети, аналіз потреб цільової аудиторії та формування концепції платформи. Головним завданням стало створення інтуїтивно зрозумілого та технічно надійного інструменту для комунікації між викладачами й учнями.
- Фаза уточнення охоплює детальне визначення функціональних вимог, моделювання архітектури системи та побудову UML-діаграм для формалізації процесів.
- Фаза конструювання включає реалізацію основних компонентів клієнтської й серверної частини, налаштування маршрутизації, взаємодії з базою даних та тестування окремих модулів.
- Фаза впровадження була спрямована на оптимізацію роботи, перевірку сумісності з різними пристроями та підготовку системи до публічного використання.[1]

Одним з найважливіших етапів проектування є формалізація ролей користувачів у системі. У даній системі передбачено дві основні ролі — адміністратора та учня. Адміністратор має повний доступ до керування платформою: перегляд і опрацювання заявок від потенційних учнів, редагування інформації про курси та вартість, оновлення візуального контенту. Учень не потребує автентифікації та може оглядати відкриту інформацію про платформу, курси та викладачів, проходити тест для визначення рівня мови, переглядати зразки навчальних матеріалів та надсилати запит на пробне заняття.

Формування чітких і детальних функціональних вимог є критично важливим етапом проектування, оскільки саме вони визначають основний можливості системи. Функціональні вимоги визначають перелік основних дій, які система має забезпечувати для кожної з типів користувачів. Ці функції формують ядро платформи – саме завдяки їм сайт перетворюється з простої презентаційної сторінки на повноцінний інструмент взаємодії між школою та користувачем[3]. В межах даного дослідження та проектування було сформовано наступні функціональні можливості платформи:

- Виведення інформації про курси з детальним описом рівнів (A1, A2, B1 тощо), тривалості, вартості та форматів занять.
- Форма запису на пробне заняття: заповнення контактних даних, вибір курсу.
- Проходження онлайн-тесту на визначення рівня володіння іспанською мовою.
- Модальна форма запису, подання заявки на курси.
- Кнопки для зв'язку з адміністрацією платформи через месенджери.
- Адміністративна панель для перегляду заявок, їх експорт або сортування.

Крім представлених можливостей, важливо, щоб платформа була надійною, зручною у використанні та технічно стабільною. Саме за це відповідають нефункціональні вимоги. Вони формують «якість» платформи з погляду користувача та розробника.

На ринку онлайн-освіти представлені такі платформи, як Duolingo, Babbel, Preply. Duolingo орієнтована на самостійне навчання через гейміфіковані вправи, але не передбачає занять з викладачем. Babbel пропонує готові уроки без можливості адаптації під український контекст. Preply дозволяє обирати викладачів, проте не має власної навчальної інфраструктури. Розроблена платформа поєднує переваги цих рішень. Вона включає інформація про структуровані курси з чітким поділом за рівнями, можливість запису на заняття з кваліфікованими викладачами, інтеграція з Zoom та Migo безпосередньо в навчальному процесі.

Іспанська мова займає особливе місце серед мов, що активно вивчаються в онлайн-форматі. Вона є однією з найпоширеніших у світі. За даними Instituto Cervantes, нею володіють понад 600 мільйонів людей, з яких близько 500 мільйонів є носіями мови. Вона є офіційною в 21 країні, включаючи Іспанію, Мексику, Аргентину, Колумбію та інші держави Латинської Америки. Іспанська входить до п'ятірки найуживаніших мов світу. Вона також є однією з шести офіційних мов Організації Об'єднаних Націй. Викладання іноземної мови онлайн має свої особливості, що впливають на структуру занять та взаємодію між викладачем і студентом. Ключовими аспектами є правильна вимова, наголоси, інтонація, усне мовлення та складні граматичні конструкції.[9]

У зв'язку з вище поданою інформацією розробка веб-платформи для ефективного надання онлайн-курсів з іспанської мови є актуальним науково-практичним завданням. Така платформа повинна враховувати особливості викладання іноземної мови онлайн, підтримувати інтерактивні інструменти та забезпечувати зручну взаємодію між студентом і викладачем. В контексті даного дослідження та розробки передбачається, що надання курсів іспанської мови відбуватиметься у форматі інтерактивного онлайн-навчання із залученням сучасних зовнішніх ресурсів, зокрема Migo (для візуальної взаємодії) та Zoom (для проведення живих онлайн-уроків). Окремо зазначається, що викладання курсів відбувається з залученням кваліфікованих викладачів, що дозволяє забезпечити високий рівень засвоєння матеріалу та індивідуальний підхід до кожного студента.

Ключовим аспектом проєктування освітніх платформ є вибір методології розробки програмного забезпечення. Для створення платформи було обрано методологію Rational Unified Process. RUP належить до ітераційних методологій і структурує життєвий цикл розробки ПЗ [4]. Основна перевага цієї методології полягає у чіткому розподілі процесу розробки на фази, кожна з яких має визначені цілі та результати. Однією з ключових особливостей є її ітераційна природа, тобто розробка відбувається не одним великим етапом, а у вигляді ітерацій. Ця методологія добре підходить для проєктів, що потребують поетапної реалізації, чіткого розуміння вимог, а також можливості поступового вдосконалення. RUP включає чотири ключові фази: початкову, уточнення, конструювання та впровадження

Нефункціональні забезпечують зручність застосування завдяки інтуїтивно зрозумілому дизайну та логічній навігації, адаптивність для портативних пристроїв, планшетів і комп'ютерів, захищеність через валідацію форм та охорону від невірних відомостей, високу швидкість завдяки оптимізації коду та зображень, а також масштабованість структури для подальшого розширення можливостей.

Враховуючи зазначені аспекти системи, наступним кроком є обґрунтування середовища розробки та технологій, що використовуватимуться під час створення продукту. Вибір середовища розробки суттєво впливає на швидкість та якість створення веб-платформ. Visual Studio Code є найпопулярнішим безкоштовним редактором для веб-розробки, який підтримує понад 50 мов програмування та має екосистему з більш ніж 30 тисяч розширень. VS Code поєднує простоту та потужність, маючи інтуїтивний інтерфейс для початківців та професійні інструменти для досвідчених програмістів.

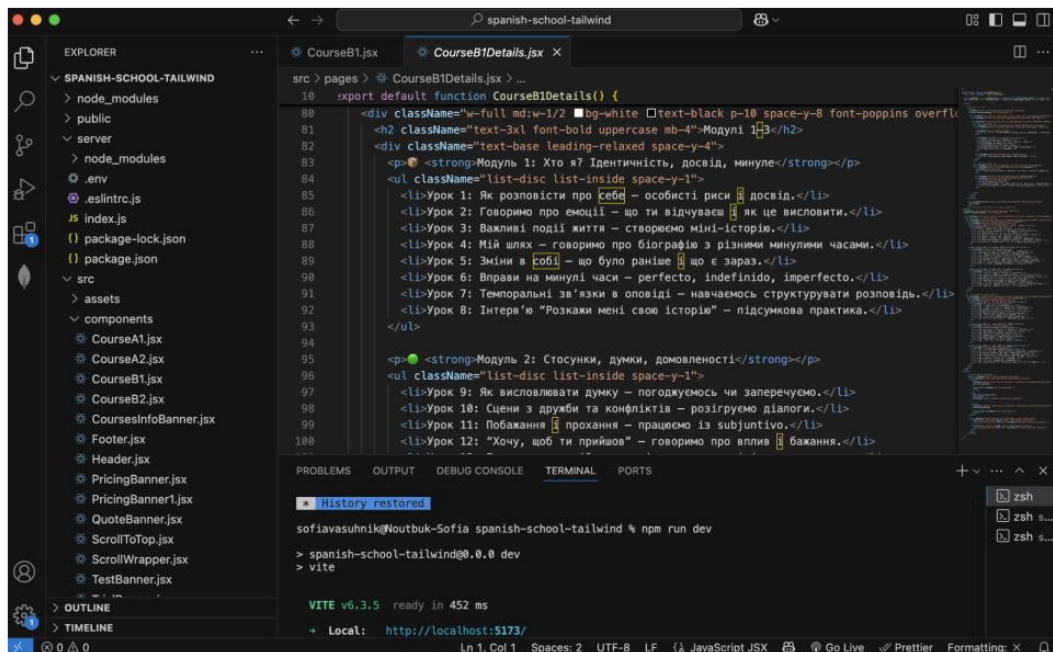


Рисунок 1 – Середовище розробки «Visual Studio Code»

Вибір технологічного стеку є стратегічно необхідним рішенням, оскільки від нього залежить продуктивність системи та швидкість розробки. Для реалізації веб-платформи використано сучасний технологічний стек Frontend та Backend, який поєднує React.js, Node.js з Express та MongoDB. Для систематизації обраних технологій та демонстрації архітектури системи було складено узагальнену таблицю технологічного стеку.

Таблиця 1 – Технологічний стек веб-платформи та його призначення

Компонент	Технологія	Призначення
Frontend	React.js	Компонентний інтерфейс
Стилізація	Tailwind CSS	Адаптивний дизайн
Маршрутизація	React Router	Навігація SPA
Backend	Node.js + Express	Серверна логіка
База даних	MongoDB	Зберігання даних
HTTP-клієнт	Axios	Комунікація клієнт-сервер

Frontend (клієнтська частина) — це видима користувачам частина веб-платформи, що включає інтерфейс, дизайн, навігацію та всі елементи взаємодії, які відображаються у браузері. Саме frontend відповідає за те, як швидко завантажуються сторінки, наскільки зручно користувач може знайти потрібну інформацію, записатися на курс чи зв'язатися зі школою. Якість frontend безпосередньо впливає на враження від платформи та ймовірність того, що відвідувач стане учнем.

Для реалізації frontend використано було обрано бібліотеку React. React дозволяє створювати інтерфейс з окремих компонентів (Header, Footer, форма запису, картки курсів) та оновлювати сторінки без перезавантаження. Tailwind CSS забезпечує адаптивний дизайн, який коректно відображається на телефонах, планшетах та комп'ютерах. React Router надає швидку навігацію між сторінками без затримок. Завдяки цьому платформа завантажується за 1-2 секунди та зручно працює на всіх пристроях.[6, 10]

Backend — це серверна частина, що є невидимою для користувача платформи. Цей компонент відповідає за обробку інформації, зберігання заявок, взаємодію з базою даних та виконання бізнес-логіки. Коли користувач заповнює форму запису на курс, саме backend отримує ці дані. Перевіривши їх коректність, зберігає у базі даних та надсилає підтвердження. Надійність backend впливає на стабільність платформи та збереження інформації про записи учнів.

Для реалізації серверної частини використано Node.js — серверне середовище, що дозволяє писати серверну логіку мовою JavaScript, як і frontend, що спрощує розробку. Фреймворк Express забезпечує швидку обробку HTTP-запитів, маршрутизацію та валідацію даних з форм. Для передачі даних між клієнтом та сервером використано бібліотеку Axios, яка надійно відправляє заявки з форми на сервер. Така архітектура забезпечує швидку обробку запитів (до 200 мс) та можливість обслуговувати сотні користувачів одночасно.[7, 11]

Для зберігання даних використовується MongoDB — хмарна база даних, яка дозволяє гнучко структурувати інформацію, надійно зберігати заявки

користувачів і легко масштабувати систему при зростанні кількості користувачів. [11]

Архітектура платформи побудована за принципом клієнт-серверної взаємодії. Клієнтська частина реалізована як односторінковий застосунок з маршрутизацією через React Router, що забезпечує швидку навігацію без перезавантаження сторінки. Серверна частина організована у вигляді RESTful API, який обробляє запити від клієнта та взаємодіє з базою даних. Комунікація між клієнтом та сервером здійснюється через бібліотеку Axios з використанням асинхронних запитів.

В контексті веброзробки ключовим елементом є створення якісного інтерфейсу, адже саме він визначає зручність взаємодії користувача із системою. Інтерфейс сайту побудований у вигляді компонентних модулів. Основні структурні компоненти веб-платформи включають такі елементи, як: «Header», «Footer», головна сторінка, розділи «Курси», «Про курс», «Ціни», інтерактивний блок «Тест на визначення рівня» та модальне вікно для запису. Кожен із модулів був розроблений як окремий файл та має свій код із логікою й оформленням. Такий підхід сприяв зручній структуризації проекту. Форма заявки містить поля для введення рівня знань, контактного телефону, нікнейму в соціальних мережах та додаткового коментаря. При успішному надсиланні заявки користувач отримує підтвердження, а дані зберігаються в базі даних.

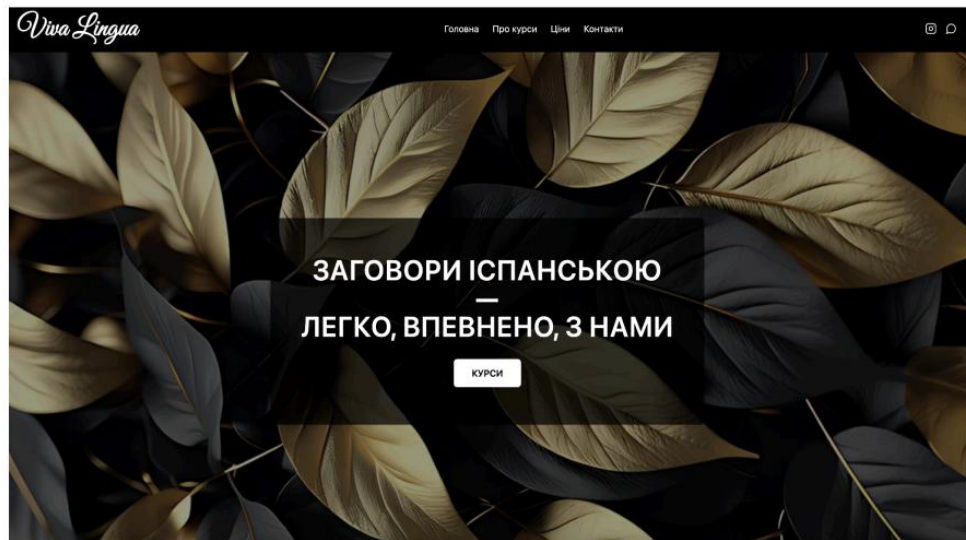


Рисунок 2 – Головна сторінка

Особливу увагу слід приділяти адаптивності інтерфейсу. Використання утилітарних класів Tailwind CSS дозволило реалізувати приємний дизайн, який коректно відображається на екрані. Для запису на курс на платформі було реалізовано модальну форму, яка відкривається при натисканні кнопки «Записатись».

Рисунок 3 – Форма для запису на курс

Тестування є важливим етапом розробки, оскільки забезпечує виявлення помилок до запуску платформи та гарантує стабільність роботи для користувачів. Для ефективного тестування корисно застосовувати інструменти Chrome DevTools для frontend-аналізу та Postman для перевірки API. В контексті даної розробки було впроваджено саме ці інструменти, а також ручне тестування форми, навігації та інтеграцій із сервісами. Завдяки комплексному підходу вдалося перевірити коректність роботи системи та усунути всі критичні помилки перед запуском.

Тестування платформи проводилося на кількох рівнях. Модульне тестування окремих компонентів React виконувалося безпосередньо під час розробки. Для інтеграційного тестування перевіряли, як клієнт взаємодіє з сервером: обробку форм, збереження даних і відображення повідомлень.

У результаті отримано повнофункціональну веб-платформу, яка демонструє ефективність використаних технологій та методологій. Система дозволяє швидко взаємодіяти, автоматизує збір заявок і представляє освітні послуги в сучасному цифровому форматі.

**Висновки.** Проведене дослідження підтверджує визначальну роль сучасних веб-технологій у розвитку онлайн-навчання, зокрема у сфері вивчення іноземних мов. У ході роботи було встановлено, що ефективність онлайн-курсів значною мірою залежить від продуманої архітектури платформи, її зручності для користувача та чіткого формування функціональних і технічних вимог.

Теоретичні аспекти, пов'язані з побудовою клієнт-серверних систем, використанням ітераційних методологій та підходів до моделювання програмних рішень, були досліджені й застосовані під час розробки структури

платформі. Разом із тим аналіз існуючих технологій дозволив обґрунтувати вибір інструментів та визначити оптимальний технологічний стек для реалізації проекту.

У результаті було досліджено, проаналізовано та впроваджено повний цикл розробки веб-платформи для надання курсів іспанської мови. Платформа виконує всі поставлені функції, дозволяючи потенційним учням отримати детальну інформацію про курси та записатися на заняття.

Проведений аналіз дозволив з'ясувати особливості побудови фронтенд- та бекенд-складових платформи, а також визначити, які технології забезпечують їх узгоджену роботу. Отримані результати дали можливість сформувати оптимальну архітектуру системи, що поєднує зручний інтерфейс, стабільну серверну логіку та ефективну взаємодію з базою даних. Застосування методології RUP забезпечило систематизацію процесу розробки та чітку структурування вимог.

Розроблена платформа може бути адаптована до реальних потреб мовних шкіл, доповнена новими модулями та служити прикладом впровадження сучасного підходу до створення освітніх веб-рішень. Перспективами подальших досліджень є впровадження системи оплати та розширення функціоналу адміністративної панелі.

#### Список використаних джерел

1. Грінченко О. А. Проектування інформаційних систем: навч. посіб. – К.: КНУ імені Тараса Шевченка, 2020. – 198 с.
2. Савінов О. Ю. Веб-програмування. Створення сучасних веб-застосунків. – Харків: Фоліо, 2021. – 304 с.
3. Бочкарьова Т. І., Степаненко І. В. Методи та засоби моделювання інформаційних систем. – Київ: КНТ, 2020. – 212 с.
4. Соммервілл І. Інженерія програмного забезпечення: навч. посіб. – 10-те вид. – К.: Діалектика, 2021. – 864 с.
5. Кайзер Г. UML-моделювання для студентів: посібник. – Львів: ЛНУ, 2020. – 156 с.
6. Браун М. React.js: Професійна розробка інтерфейсів. – К.: Нове знання, 2022. – 320 с.
7. Браунлі Дж. Node.js на практиці: створення масштабованих веб-додатків. – Харків: Видавництво «Фактор», 2021. – 276 с.
8. Сидоренко Л. С., Мельничук О. В. Інформаційні системи в економіці та бізнесі: навч. посіб. – Київ: Центр учбової літератури, 2021. – 220 с.
9. Європейська мовна політика і цифрова освіта: перспективи вивчення іспанської мови в онлайн-середовищі / наук. ред. Т. В. Шевченко. – Київ: Наук. думка, 2022. – 186 с.
10. React Documentation. React 18 [Електронний ресурс]. – Режим доступу: <https://react.dev/learn> (дата звернення: 30.08.2025).
11. Node.js Documentation [Електронний ресурс]. – Режим доступу: <https://nodejs.org/en/docs> (дата звернення: 10.09.2025).
12. Rational Unified Process. IBM Knowledge Center [Електронний ресурс]. – Режим доступу: <https://www.ibm.com/docs/en/rup/> (дата звернення: 19.09.2025)

## Додаток Б

В додатку Б наведено лістинг коду клієнтського компонента форми подання заявки на курс

Лістинг 2.1 – Фрагмент коду клієнтського компонента форми подання заявки на курс

---

```
{showModal && (
  <div className="fixed inset-0 bg-black bg-opacity-80 flex justify-center items-center z-50">
    <div className="bg-black border-2 border-[#f1c59a] p-10 rounded-xl w-[500px]">
      <h2 className="text-5xl font-kitsch text-[#f1c59a] text-center mb-8">Запис на курс
    </h2>
    {successMessage ? (
      <div className="text-center text-white text-2xl font-kitsch">
        <p>{successMessage}</p>
      </div>
    ) : (
      <form onSubmit={handleSubmit} className="space-y-6">
        <input
          type="text"
          name="name"
          placeholder="Ім'я / нікнейм"
          value={formData.name}
          onChange={handleChange}
          className="w-full p-4 text-black rounded-xl text-xl"
          required
        />
        <input
          type="tel"
          name="phone"
          placeholder="Номер телефону"
          value={formData.phone}
          onChange={handleChange}
          className="w-full p-4 text-black rounded-xl text-xl"
          required
        />
        <input
          type="email"
          name="email"
          placeholder="Email"
          value={formData.email}
          onChange={handleChange}
          className="w-full p-4 text-black rounded-xl text-xl"
          required
        />
        <textarea
          name="comment"
```

```

placeholder="Коментар (необов'язково)"
value={formData.comment}
onChange={handleChange}
rows={3}
className="w-full p-4 text-black rounded-xl text-xl"
/>
<div className="flex justify-between mt-6">
  <button type="submit" className="bg-[#f1c59a] text-black px-6 py-3 rounded-xl
text-xl font-bold">
    Надіслати
  </button>
  <button
    onClick={() => setShowModal(false)}
    type="button"
    className="text-white text-xl underline"
  >
    Скасувати
  </button>
</div>
</form>
)}
</div>
</div>
)}

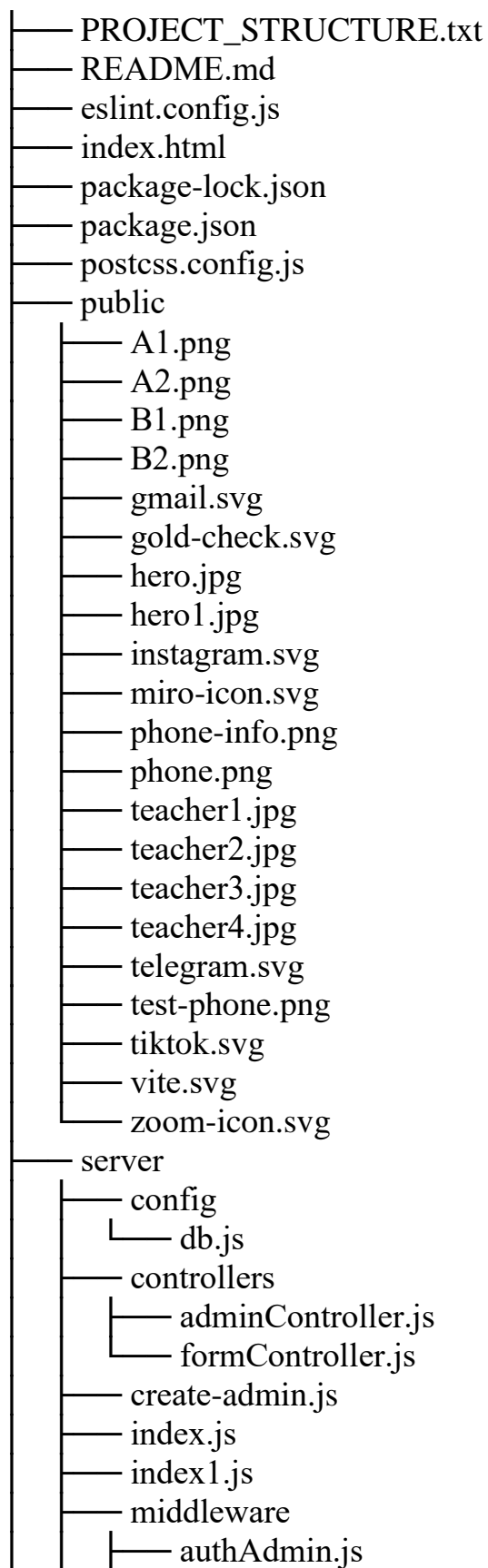
```

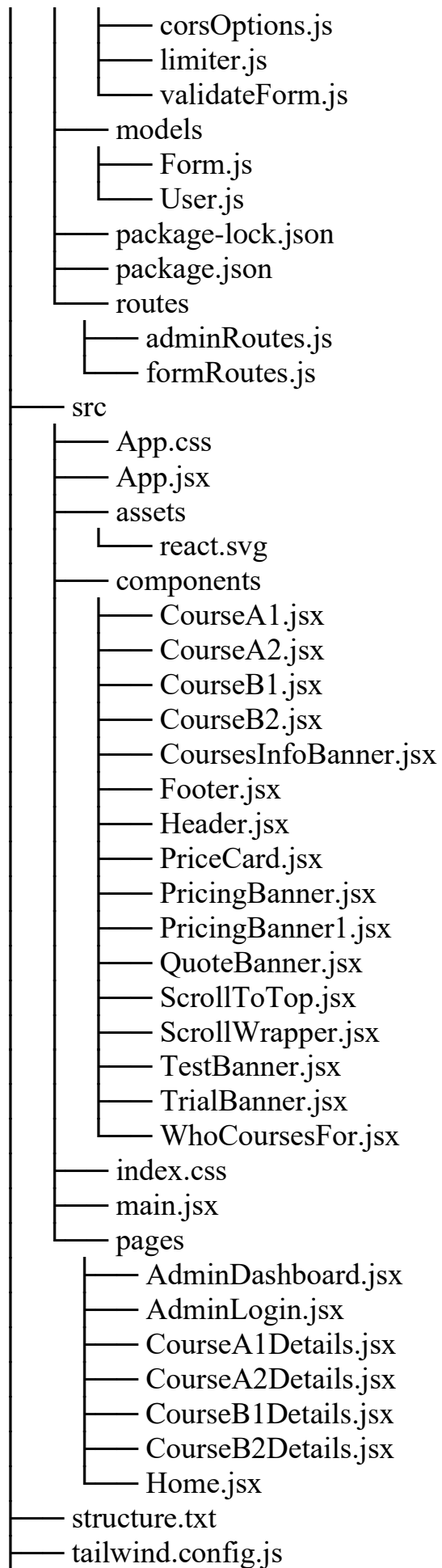
---

кінець лістингу 2.1

## Додаток В

В додатку В подано структуру директорій веб-платформи.





└─ vite.config.js

12 directories, 75 files