

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

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

**РОЗРОБКА ОНЛАЙН-ПЛАТФОРМИ ДЛЯ ОСВІТНЬОГО ПРОЦЕСУ З
ВИКОРИСТАННЯМ JAVASCRIPT ТА PHP**

**DEVELOPMENT OF AN ONLINE PLATFORM FOR THE EDUCATIONAL
PROCESS USING JAVASCRIPT AND PHP**

спеціальність 121 «Інженерія програмного забезпечення»
освітня програма «Інженерія програмного забезпечення»

Виконала: здобувач вищої освіти
групи ІПЗ-42
Чоботар Юлія Сергіївна

Керівник:
Вознюк Анастасія Вадимівна

Кваліфікаційну роботу
допущено до захисту

«10» 06 2025 р.

Гарант освітньої програми:

к.т.н., доцент

Ліщина Наталія Миколаївна



Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри


«20» 12 2024р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Чоботар Юлії Сергіївні

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи «Розробка онлайн-платформи для освітнього процесу з використанням JavaScript PHP»

Керівник роботи: асистент Вознюк А. В.

затверджені наказом закладу вищої освіти від «19» грудня 2024 р. № 474/01-02

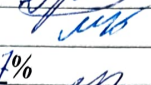
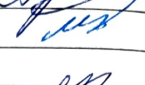
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи бакалавра «10» червня 2025 р.

3. Вихідні дані до роботи: HTML, CSS, JavaScript, PHP, MySQL, XAMPP, VS Code, методичні вказівки до виконання кваліфікаційної роботи бакалавра

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити): проаналізувати аналоги платформ, сформулювати вимоги до системи, змодельювати UML-діаграми, обґрунтувати вибір технологій, реалізувати вебплатформу, провести тестування, впровадити захист та оптимізувати пошукову систему для онлайн-платформи

5. Перелік графічного матеріалу: 45 рисунків, 1 додаток

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Вознюк А. В.		
Специфікація вимог до розробленої системи	Вознюк А. В.		
Розробка об'єкта проєктування	Вознюк А. В.		
Нормоконтроль	Вознюк А. В.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту		101%	
Академічна доброчесність	Вознюк А. В.		

7. Дата видачі завдання «20» грудня 2024 р.

№	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	до 04.02.2025 р.	Виконано
2	Аналіз проблеми розробки та впровадження об'єкту проєктування	до 01.03.2025 р.	Виконано
3	Обґрунтування вибору шляхів, технологій і засобів вирішення поставленого завдання	до 15.03.2025 р.	Виконано
4	Розробка функціонально-структурної схеми роботи об'єкта проєктування та проєктування бази даних	до 29.03.2025 р.	Виконано
5	Практична реалізація об'єкта проєктування та розробка бази даних	до 26.04.2025 р.	Виконано
6	Тестування та налагодження об'єкта проєктування	до 03.05.2025 р.	Виконано
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедрі	до 10.06.2025 р.	Виконано

Здобувач вищої освіти



Юлія ЧОБОТАР

Керівник кваліфікаційної роботи



Анастасія ВОЗНЮК

АНОТАЦІЯ

Чоботар Ю. С. Розробка онлайн-платформи для освітнього процесу з використанням JavaScript та PHP. Рукопис. Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення» спеціальності «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2025.

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків.

У першому розділі здійснено аналіз сучасного стану онлайн-освіти, розглянуто провідні освітні платформи, виявлено їхні переваги й недоліки, а також обґрунтовано доцільність створення власної онлайн-платформи. Сформульовані завдання кваліфікаційної роботи. У другому розділі наведено специфікацію вимог до програмного забезпечення, визначено архітектуру системи, змодельовано функціональні сценарії роботи платформи із використанням UML-діаграм, обґрунтовано вибір технологій розробки. У третьому розділі реалізовано практичну частину: створено інтерфейси користувача та адміністратора, базу даних, функціонал реєстрації, авторизації, створення курсів, коментування, вподобань, пошуку. Проведено тестування та налагодження онлайн-платформи, впроваджено заходи інформаційної безпеки та оптимізовано онлайн-платформу для пошукових систем.

У висновках підсумовано результати реалізації онлайн-платформи, оцінено її відповідність поставленим вимогам і функціональність.

Ключові слова: онлайн-платформа, PHP, JavaScript, база даних, навчання, веброзробка, UML-діаграма, тестування, інтерфейс.

ABSTRACT

Chobotar Y. Development of an online platform for the educational process using JavaScript and PHP. Manuscript. Qualification work of the bachelor's program "Software engineering" in the specialty "Software engineering". Lutsk National Technical University. Lutsk, 2025.

The bachelor's qualification thesis consists of an introduction, three chapters, conclusions, a list of references, and appendices.

The first chapter analyzes the current state of online education, examines the leading educational platforms, identifies their advantages and disadvantages, and substantiates the feasibility of creating your own online platform. The tasks of the qualification work are formulated. The second section provides a specification of software requirements, defines the system architecture, models functional scenarios of the platform using UML diagrams, and justifies the choice of development technologies. The third section describes the practical part of the project: user and administrator interfaces, a database, registration, authorization, course creation, commenting, preferences, and search functionality. The online platform was tested and debugged, information security measures were implemented, and the online platform was optimized for search engines.

The conclusions summarize the results of the implementation of the online platform, evaluate its compliance with the requirements and functionality, and outline the prospects for further development of the system.

Keywords: online platform, PHP, JavaScript, database, training, web development, UML diagram, testing, interface.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ	9
1.1 Аналіз сучасного стану проблеми	9
1.2 Постановка завдання на кваліфікаційну роботу бакалавра	17
Висновки по розділу 1	18
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ	19
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проєктування програмного забезпечення	19
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання	28
Висновки по розділу 2	30
РОЗДІЛ 3 РОЗРОБКА ОНЛАЙН-ПЛАТФОРМИ ДЛЯ ОСВІТНЬОГО ПРОЦЕСУ З ВИКОРИСТАННЯМ JAVASCRIPT ТА PHP	32
3.1 Практична реалізація об'єкта проєктування	32
3.2 Тестування та налагодження інформаційно-комп'ютерної системи.....	46
3.3 Проєктування структури бази даних.....	50
3.4 Захист інформаційно-комп'ютерної системи.....	57
3.5 SEO інформаційно-комп'ютерної системи	58
Висновки по розділу 3	59
ВИСНОВКИ.....	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	63
ДОДАТКИ.....	65

ВСТУП

Актуальність теми. У сучасних умовах активного впровадження цифрових технологій онлайн-навчання стало невіддільною частиною освітнього процесу. Після пандемії COVID-19 попит на зручні, доступні та функціональні вебплатформи суттєво зріс. Водночас більшість наявних платформ мають низку обмежень – складний інтерфейс, відсутність аналітики, обмежену персоналізацію та взаємодію з контентом. Це створює потребу у створенні більш доступних та інтуїтивних платформ, які поєднували б технічну надійність і освітню ефективність. Тому розробка сучасної вебплатформи для онлайн-освіти, яка враховує потреби як користувачів, так і адміністраторів, є актуальним і практично значущим завданням.

Таким чином, метою даної кваліфікаційної роботи є розробка функціональної, безпечної та безкоштовної онлайн-платформи для організації онлайн-навчання з можливістю взаємодії між студентами й адміністраторами, управління контентом, комунікації та аналітики.

Об'єктом кваліфікаційної роботи є онлайн-платформа. Вона має на меті сприяти зручному доступу до навчального матеріалу для студентів, які навчаються дистанційно та у разі непередбачуваних обставин коли студент, не зможе бути присутнім на парі він зміг ознайомитись з матеріалом.

Предметом кваліфікаційної роботи є практична реалізація онлайн-платформи використовуючи JavaScript та PHP.

Для досягнення поставленої мети були поставлені наступні завдання:

- провести огляд предметної області та проаналізувати наявні рішення у сфері онлайн-навчання;
- сформулювати функціональні та нефункціональні вимоги до платформи;
- побудувати UML-діаграми для моделювання логіки кваліфікаційної роботи;

– обґрунтувати та обрати мови програмування, СКБД і середовища розробки;

- реалізувати вебплатформу з відповідним функціоналом;
- протестувати платформу, виявити та усунути помилки;
- впровадити базові механізми інформаційного захисту;

З метою апробації матеріалів роботи підготовлено тези, які були опубліковані в збірнику X Міжнародної науково-практичної конференції з проблем вищої освіти і науки «Інформаційні технології в освіті, науці і виробництві (ІТОНВ-2025)» м. Луцьк, 23-24 травня 2025 р. (додаток А).

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

До 2020 року онлайн-освіта залишалася для багатьох людей другорядною альтернативою традиційному навчанню. Курси та платформи існували, але в більшості вони сприймалися як доповнення до освітнього процесу. Однак пандемія COVID-19 послужила каталізатором для швидкого адаптування до нових умов, і дистанційне навчання стало не просто вимушеною необхідністю, а й можливістю.

З цього моменту онлайн-освіта почала активно розвиватися. Мільйони людей по всьому світу відкрили для себе зручність нових форматів: більше не було потреби витратити час на дорогу, підлаштовуватися під розклади чи місця проведення занять; достатньо було мати доступ до інтернету – і знання були під рукою.

Підкріплює це і статистика: у 2019 році лише 36,3 % студентів у США проходили хоча б один онлайн-курс, а вже у 2020 ця цифра стрімко зросла до понад 75 % [1]. Зростання зацікавлення онлайн-освітою видно й у Google Trends – під час першого локдауну запити на тему «online learning» злетіли в десятки разів (рис. 1.1).

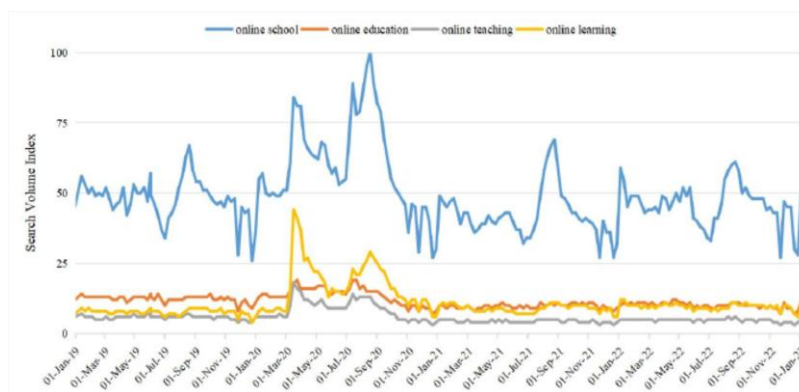


Рисунок 1.1 – Графік пошукових запитів 2019-2023 років [1]

За даними World Economic Forum, інвестиції у сферу EdTech досягли \$ 18,66 млрд ще до пандемії, до 2025 року очікується зростання до \$ 350 млрд. Це свідчить не лише про короточасний сплеск, а про формування нової норми у сфері освіти [2].

Онлайн-освіта довела свою ефективність, адаптивність і здатність охопити найрізноманітніші аудиторії. Вона відкриває доступ до знань незалежно від місця проживання, соціального статусу чи віку, і дає змогу навчатися у власному ритмі.

В Україні пандемія коронавірусної хвороби також стала поштовхом до тотального переходу на дистанційне навчання. За даними Міністерства освіти і науки України, у 2020-2021 роках доступність освіти зросла в п'ять разів порівняно з 2016-2017 роками, завдяки впровадженню онлайн-форматів [3].

Опитування Державної служби якості освіти України у квітні 2021 року показало, що лише 49,5 % респондентів бажають повернення до повністю офлайн-навчання після карантину, тоді як 8 % вже надають перевагу онлайн-формату – це вдвічі більше, ніж у 2020 році. Дослідження, проведене серед 882 студентів і викладачів з 65 українських вишів, виявило, що хоча перехід до онлайн-навчання був вимушеним і супроводжувався труднощами, він також відкрив нові можливості для цифровізації освіти та розвитку цифрових навичок [4].

Загалом, пандемія стала рушійною силою зростання онлайн-навчання в Україні, полегшивши впровадження новітніх інструментів і методик у навчальні програми. Онлайн-навчання в Україні перетворилося з альтернативи на необхідність, а потім стало важливою частиною освітньої системи, як і в глобальному контексті. Її розвиток продовжується, відкриваючи нові можливості для доступу до знань та підвищення кваліфікації населення. Станом на 2025 рік онлайн-освіта є актуальною як в Україні, так і в багатьох інших країнах світу. Вона давно перестала бути тимчасовим рішенням, запровадженим під час пандемії COVID-19, і перетворилася на стійку форму отримання знань, яка активно використовується у формальній, неформальній та післядипломній

освіті. Доступність і зручність, що дозволяє студентам обирати час, місце і темп навчання, пояснюють його популярність. Це особливо важливо для тих, хто живе у віддалених районах, має обмежену мобільність або поєднує роботу з навчанням.

Технологічний прогрес значно розширив можливості онлайн-освіти: впровадження штучного інтелекту, віртуальної та доповненої реальності, автоматизованих систем оцінювання та персоналізованого навчального контенту зробило цей формат ще більш ефективним. Крім того, онлайн-курси набули широкого поширення серед дорослого населення, особливо в контексті підвищення кваліфікації, перекваліфікації та здобуття нових професійних навичок, що відповідають сучасним вимогам ринку праці. Сприяє популярності й відносна економічність дистанційного навчання – користувачам не потрібно витрачати кошти на проїзд, проживання або оренду навчальних приміщень.

Онлайн-навчання має недоліки, незважаючи на свої численні переваги. Одним з основних недоліків є відсутність живого, безпосереднього спілкування, що може вплинути на мотивацію та соціалізацію студентів. Крім того, якість навчального контенту на різних платформах може суттєво відрізнятися, і не всі практичні навички можна повноцінно розвивати у віддаленому форматі. Цифровий розрив все ще залишається значною проблемою, оскільки не всі користувачі мають достатній доступ до технічних ресурсів та швидкого Інтернету, що особливо помітно в регіонах з недостатньо розвинутою інфраструктурою.

В Україні зберігається сталий інтерес до онлайн-освіти. Після завершення суворих карантинних обмежень чимало закладів вищої освіти залишили елементи дистанційного формату у вигляді гібридного навчання. Водночас активно розвивається сегмент неформальної освіти: все більше громадян користуються онлайн-платформами для самостійного опанування нових тем, зокрема через курси Prometheus, EdEra, Дія.Освіта, Kwiga та Coursera.

Prometheus – це популярна українська платформа масових відкритих онлайн-курсів, яка пропонує безкоштовний доступ до якісної освіти від

провідних викладачів і університетів України. Її місія – надати безкоштовний доступ до якісної освіти широкому колу громадян, включно зі школярами, студентами, держслужбовцями, ветеранами та представниками громадського сектору. Платформа тісно співпрацює з українськими вишами, експертами, міжнародними донорами й державними структурами [5].

Особливістю Prometheus є зосередженість на актуальних соціально важливих темах: цифрова та медійна грамотність, публічне управління, фінансова обізнаність, державна безпека, медицина, англійська мова, ІТ-навички тощо. Інтерфейс платформи простий, зрозумілий, мінімалістичний, із-за чого вона є зручною для широкого кола користувачів, навіть без досвіду використання онлайн-платформ. Він підтримує формат відеолекцій, тестування, видачу сертифікатів після проходження курсу, має форум для обговорення (рис. 1.2).

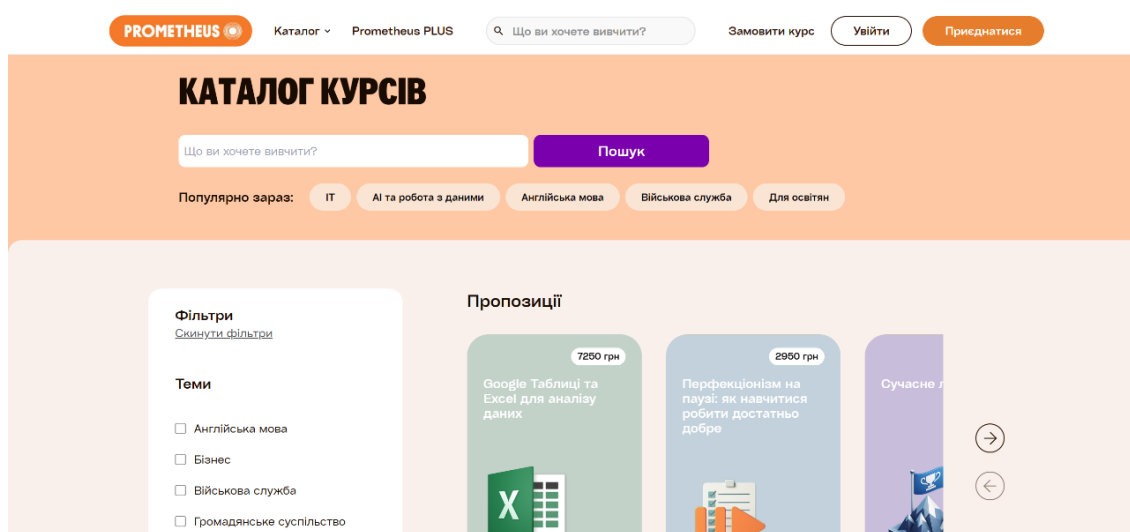


Рисунок 1.2 – Освітня платформа Prometheus [5]

Проте платформа обмежена в технічному функціоналі: відсутні адаптивні рекомендації, мобільний застосунок, автоматизована аналітика прогресу користувача. Адмінінтерфейс теж мінімальний і не дозволяє гнучко керувати курсами чи студентами.

EdEra – це українська онлайн-платформа, орієнтована на розвиток шкільної та загальної освіти через використання сучасних цифрових технологій. Вона активно співпрацює з Міністерством освіти і науки України, закладами

вищої освіти та міжнародними організаціями, що забезпечує високий рівень якості навчального контенту [6].

Платформа пропонує інтерактивні курси, електронні підручники, відеолекції та практичні завдання з широкого спектра дисциплін, включно з математикою, історією, біологією, українською та англійською мовами. Значна увага приділяється підтримці вчителів: створюються методичні матеріали, гідні та рекомендації.

Інтерфейс EdEra зручний, сучасний і адаптований під потреби як студентів, так і педагогів (рис. 1.3).

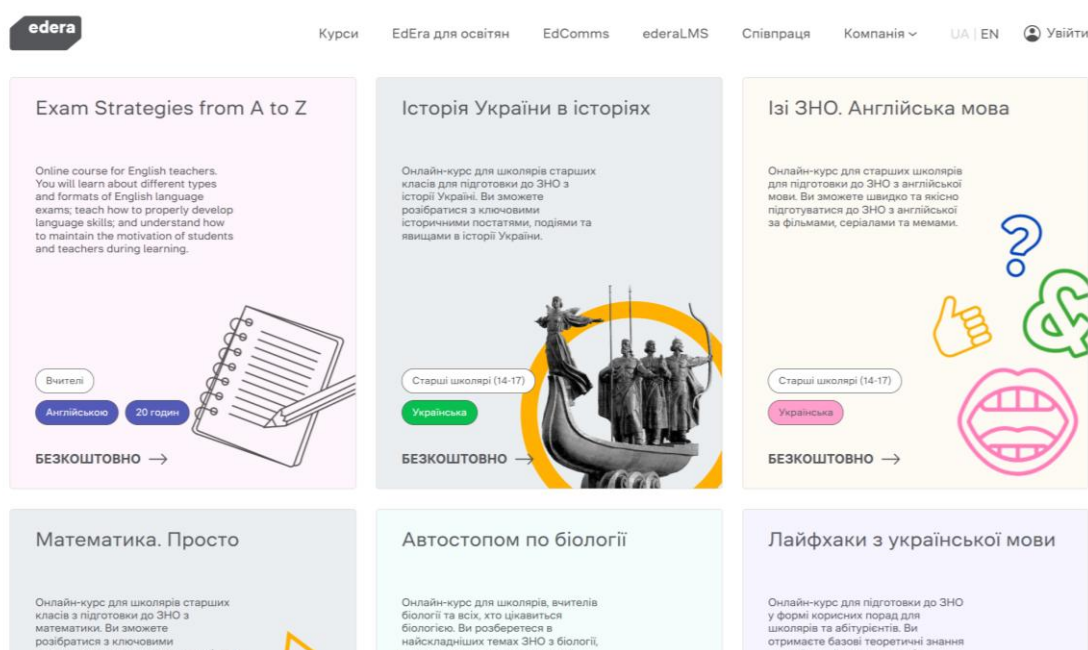


Рисунок 1.3 – Освітня платформа EdEra [6]

Платформа також реалізує спеціальні освітні проєкти, зокрема з прав людини, критичного мислення, енергетики.

Популярності набули державні й приватні ініціативи, спрямовані на підвищення цифрової грамотності та розширення доступу до якісних освітніх матеріалів.

Дія.Освіта – державна освітня платформа з короткими інтерактивними курсами, спрямованими на розвиток цифрової грамотності та актуальних навичок для громадян усіх вікових категорій. Ініціативу реалізує Міністерство

цифрової трансформації у співпраці з міжнародними партнерами та українськими експертами [7].

Ключовим форматом навчання є освітні серіали – короткі інтерактивні відео, що подають інформацію в легкій, доступній формі. Темі охоплюють базові цифрові навички, кібербезпеку, медіаграмотність, онлайн-послуги, роботу в соціальних мережах тощо.

Платформа орієнтована на широке коло користувачів, зокрема літніх людей, державних службовців, переселенців, освітян. Після проходження курсів користувачі можуть отримати сертифікати.

Дія.Освіта має зручний мобільний адаптивний інтерфейс, проте обмежена у функціональності глибокої аналітики, персоналізованих рекомендацій та розгалужених навчальних програм (рис. 1.4).



Рисунок 1.4 – Освітня платформа Дія.Освіта [7]

Kwiga – це сучасна українська універсальна платформа для створення та проведення онлайн-курсів, вебінарів і тренінгів, орієнтована на викладачів і бізнес, з широкими можливостями для монетизації та автоматизації навчального процесу з і зручним адаптивним інтерфейсом, який зображений на рисунку 1.5 [8].

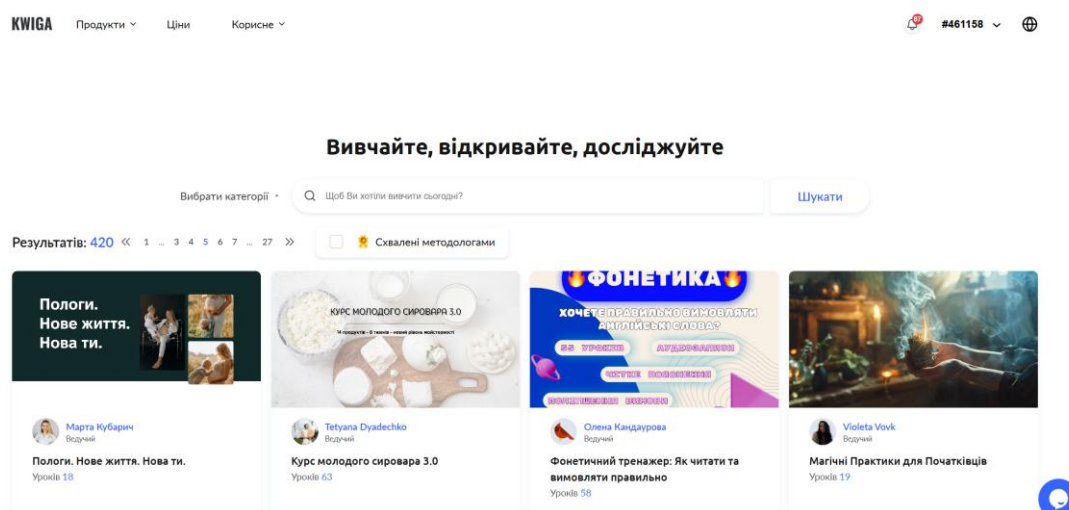


Рисунок 1.5 – Освітня платформа KWIGA [8]

Основна цільова аудиторія – експерти, коучі, власники онлайн-шкіл і маркетологи, які шукають інструмент для монетизації знань. Платформа дозволяє створювати багатокомпонентні курси з використанням таких форматів, як відео-уроки, аудіофайли, текстові матеріали, презентації, інтерактивні тести, вебінари, а також завдання з перевіркою вручну.

Kwiga надає розвинену систему керування доступом: можна налаштувати прогресивне відкриття модулів, обмеження за часом, обмеження кількості переглядів тощо. Також передбачено гнучку сегментацію аудиторії, що дозволяє персоналізувати навчання. Додатково реалізовано інтеграції з CRM, аналітичними сервісами, маркетинговими інструментами, email-розсилки, промокоди, бонусні програми, платіжними шлюзами та чат-ботами.

Серед недоліків слід виділити складність для новачків через велику кількість налаштувань, переобтяженість інтерфейсу в адмінпанелі, а також суттєві обмеження безкоштовного тарифного плану – більшість корисних функцій доступні лише на платній основі. Це обмежує використання платформи для неприбуткових навчальних ініціатив.

Coursera – одна з найвідоміших міжнародних платформ для онлайн-навчання, яка працює на ринку з 2012 року та надає доступ до курсів, спеціалізацій і сертифікатів від університетів та компаній з усього світу, з можливістю навчання у зручному темпі [9]. Її особливістю є партнерство з

провідними університетами, таких як Stanford, Yale та University of London, і світовими компаніями, таких як Google, IBM, Meta, що дозволяє пропонувати тисячі високоякісних курсів, програм сертифікації, спеціалізацій і навіть повноцінних онлайн-ступенів (рис. 1.6).

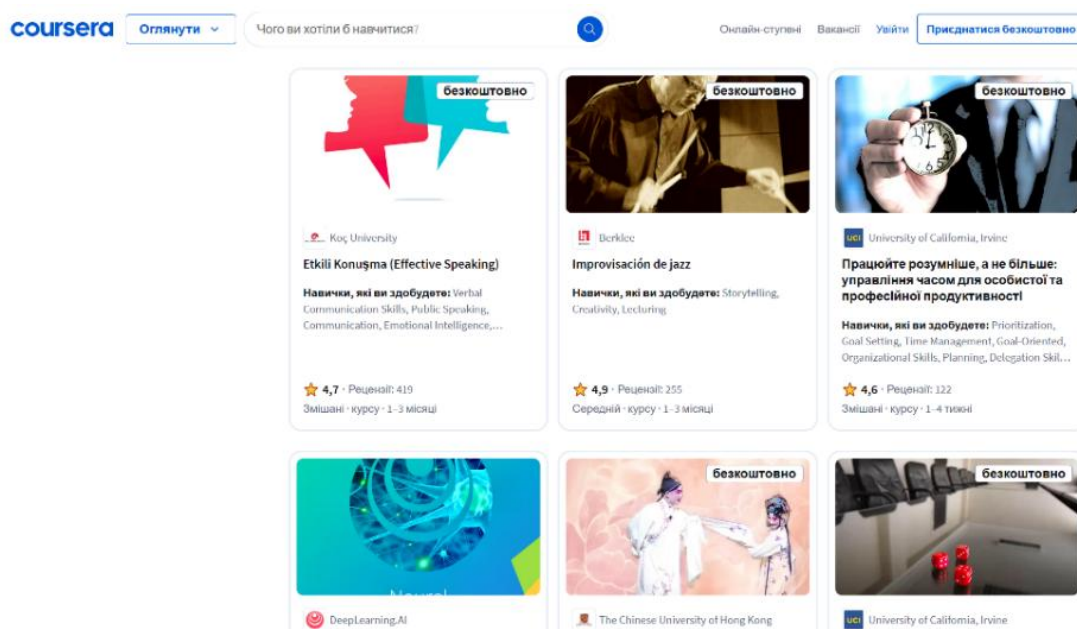


Рисунок 1.6 – Освітня платформа COURSERA [9]

Курси на Coursera мають чітку структуру, розділену на тижні, з відеолекціями, проміжними тестами, фінальними проєктами та інтерактивними завданнями. На ній вбудовано систему автоматичного оцінювання, peer-review, гейміфікації та форумів. Також вона підтримує адаптацію під різні мови, має мобільний застосунок, систему рекомендацій на основі інтересів та вподобань користувача.

Недоліками Coursera є високий рівень складності для нових користувачів, особливо тих, хто не володіє англійською мовою на достатньому рівні, а також можливість додавання курсів – лише затверджені партнери можуть публікувати освітній контент. Безкоштовна версія обмежує доступ до завдань і сертифікатів, що знижує цінність для багатьох студентів. Більшість інтерактивних можливостей – комунікація, наставництво, індивідуальні поради – доступні тільки в платних програмах.

У ході аналізу аналогів було виявлено потребу в розробці більш доступної платформи, так як багато з них мають обмеження щодо безкоштовного доступу, яка б сприяла як якісному управлінню освітнім контентом, так і всебічній взаємодії користувачів з платформою. Майбутня платформа передбачає два типи користувачів – адміністратора та студента, для кожного з яких буде реалізовано окремий функціонал.

Функціональність для користувача передбачатиме перегляд активних навчальних плейлистів та матеріалів, можливість взаємодіяти з ним, переглядати особистий профіль з аналітикою – кількість збережених плейлистів, вподобаних відео, написаних коментарів, можливість переходу безпосередньо до збереженого або вподобаного контенту, пошук за курсами та викладачами.

З боку адміністратора, він матиме розширений набір можливостей, який включатиме створення та редагування навчальних плейлистів та контенту, перегляд повного списку створеного контенту та усіх коментарів залишеними студентами та можливість їх видаляти, також він матиме змогу переглядати інформацію профілю з відображенням статистики: кількістю створених плейлистів та матеріалів, загальна кількість коментарів із зазначенням студента, який його написав.

Таким чином, майбутня платформа поєднує функціональні можливості сучасних навчальних платформ з додатковими перевагами – глибшою інтеграцією з профілем студента, а також розширеною статистикою для адміністраторів.

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

Для досягнення поставленої мети розробки онлайн-платформи для освітнього процесу з використанням JavaScript та PHP необхідно виконати такі завдання:

- здійснити аналіз предметної області онлайн-освіти та сучасного стану проблеми. Проаналізувати наявні аналоги, оцінити їх функціональні можливості, недоліки;
- на основі проведеного аналізу сформулювати функціональні та нефункціональні вимоги до розроблюваної онлайн-платформи;
- розробити UML-діаграми, які ілюструють логіку роботи онлайн-платформи;
- обрати засоби розробки, а саме мови програмування для серверної та клієнтської частини, систему управління базами даних, а також середовище локального розгортання;
- розробити онлайн-платформу для освітнього процесу відповідно до сформульованих вимог;
- здійснити тестування працездатності онлайн-платформи, забезпечити стабільність роботи;
- спроектувати структуру бази даних;
- впровадити заходи інформаційної безпеки онлайн-платформи;
- оптимізувати онлайн-платформу для пошукових систем.

Висновки по розділу 1

У першому розділі було здійснено аналіз сучасного стану розвитку онлайн-освіти, проведено аналіз відомих мені онлайн-платформи та проведено детальний огляд, який допоміг виявити їхні переваги та недоліки. На основі виявлених особливостей було сформовано ідею створення власної онлайн-платформи, яка прагне запропонувати альтернативу з новими підходами до зручності та взаємодії. Доцільність розробки обумовлена бажанням реалізувати низку цікавих функцій як для користувачів, так і для адміністраторів. На основі проведеного аналізу сформульовано перелік завдань, реалізація яких дозволить досягнути поставленої мети кваліфікаційної роботи.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проєктування програмного забезпечення

Наступним етапом після аналізу вже наявних платформ, було визначення функцій вимог, які мають бути реалізовані в системі для двох основних типів користувачів – звичайного користувача в нашому випадку студента та адміністратора.

Зокрема, була побудована діаграма варіантів використання, для візуального представлення функціональних можливостей онлайн-платформи (рис. 2.1). А саме функціональні можливості онлайн-платформи для студента та адміністратора.

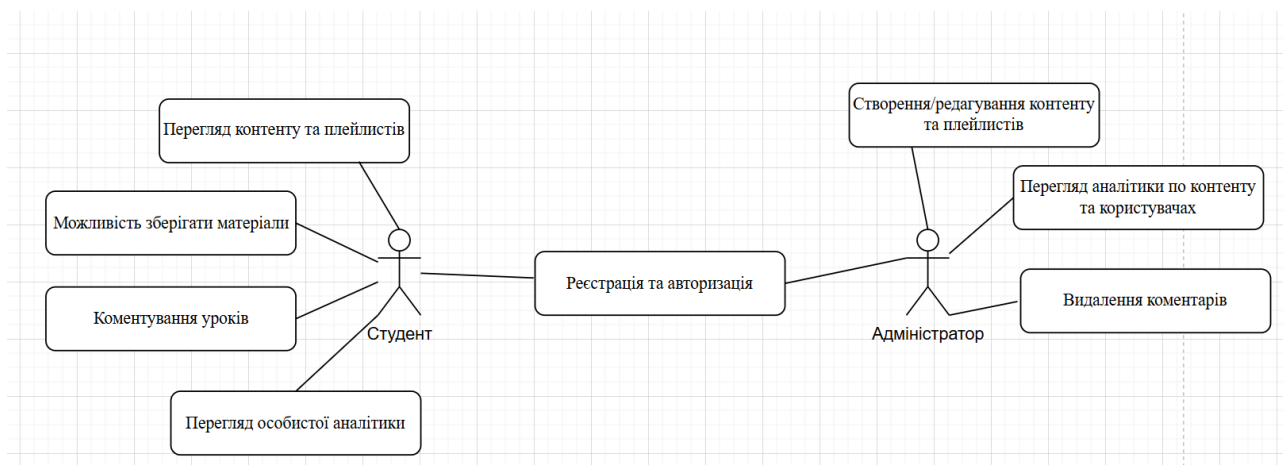


Рисунок 2.1 – Діаграма використання

Функціональні вимоги для студента:

- реєстрація та авторизація користувача, що надає можливість зареєструватись або увійти в новий або вже наявний акаунт;
- перегляд контенту та плейлистів, користувач може переглядати доступний навчальний контент;

- можливість додавати вподобання, зберігати матеріали, це дозволить користувачу формувати персоналізовану бібліотеку навчальних матеріалів;
- коментування уроків, стимулювати взаємодію між користувачами та зворотний зв'язок для викладачів;
- перегляд особистої аналітики, щоб забезпечити користувача коротким оглядом взаємодії з платформою.

Функціональні вимоги для адміністратора, тобто викладача:

- реєстрація та авторизація викладача, що надає можливість зареєструватись або увійти в новий або вже наявний акаунт;
- додавання нового контенту, редагування або видалення існуючого, тобто надати адміністратору повний контроль над навчальним контентом;
- створення плейлистів, структурувати навчальні матеріали у логічні послідовності для користувачів;
- перегляд аналітики по контенту та користувачах, що надає адміністратору деталізовану інформацію про взаємодію користувачів з контентом;
- модерація коментарів (видалення неприйнятної змісту), щоб забезпечити безпечне та конструктивне середовище для користувачів.

Також були сформульовані нефункціональні вимоги, які важливі для якості роботи платформи:

- інтерфейс платформи має бути інтуїтивно зрозумілою як для нових користувачів, так і для адміністратора;
- цілісність бази даних, тобто всі операції виконуються з урахуванням зв'язків між таблицями, уникнення помилок при введенні/видаленні даних;
- розмежування прав доступу, а саме звичайні користувачі не мають доступу до функцій адміністратора, що реалізовано на рівні логіки сайту.

На основі зібраних вимог я сформувала уявлення про загальну архітектуру майбутнього програмного продукту. Для наочного представлення структури та логіки взаємодії компонентів системи було застосовано структурно-об'єктну

модель із використанням нотації UML (Unified Modeling Language) – стандартної мови моделювання програмного забезпечення.

Визначивши функціональні вимоги, я перейшла до конкретизації того, які саме дані повинні бути присутні в системі для підтримки кожної функції.

Для цього була розроблена узагальнена діаграма (рис. 2.2), яка демонструє структуру даних усіх ключових об'єктів. Вона стала основою для побудови бази даних і логіки роботи інтерфейсу.

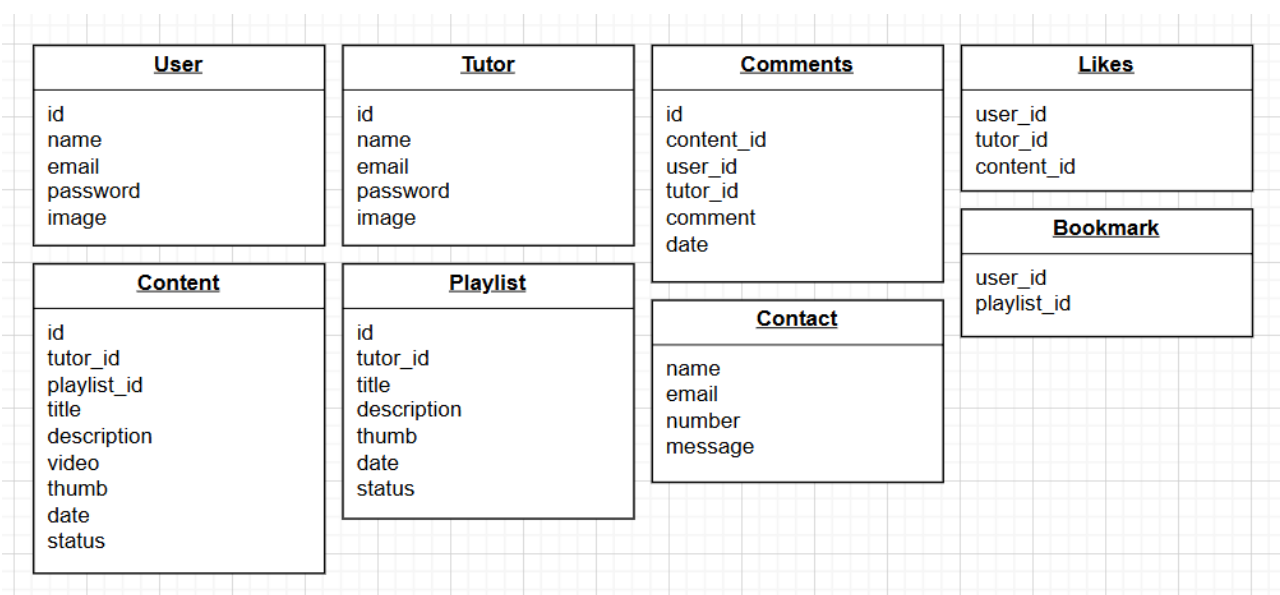


Рисунок 2.2 – Діаграма структури даних системи

Для ефективної реалізації логіки системи було використано не лише моделювання у нотації UML, а й зокрема побудовано діаграми активності, які демонструють покрокову взаємодію певного актора з платформою – від ініціації дії до отримання результату.

Для відображення логіки функціоналу перегляду профілю було створено діаграму (рис. 2.3), яка ілюструє ключові етапи взаємодії користувача із платформою – від ініціації запиту до отримання та виведення персоналізованої інформації на екран. Діаграма демонструє, як платформа звертається до бази даних, витягує дані про активність користувача, після чого оброблена інформація виводиться у вигляді аналітики.

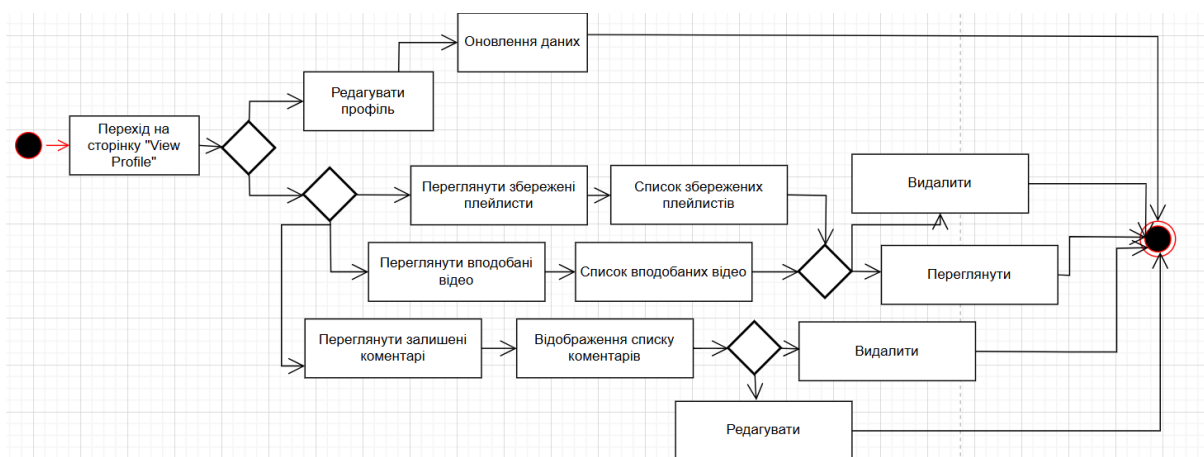


Рисунок 2.3 – UML-діаграма активності: перегляд профілю користувачем

З метою наочного відображення внутрішньої логіки процесу публікації коментаря користувачем було здійснено моделювання послідовності дій, що супроводжують виконання цієї операції в межах функціоналу вебплатформи. Такий підхід дозволяє детально проаналізувати усі етапи взаємодії між користувачем, клієнтським інтерфейсом та серверною частиною системи. Попередньо проведено аналіз послідовності, які зображенні на рисунку 2.4.

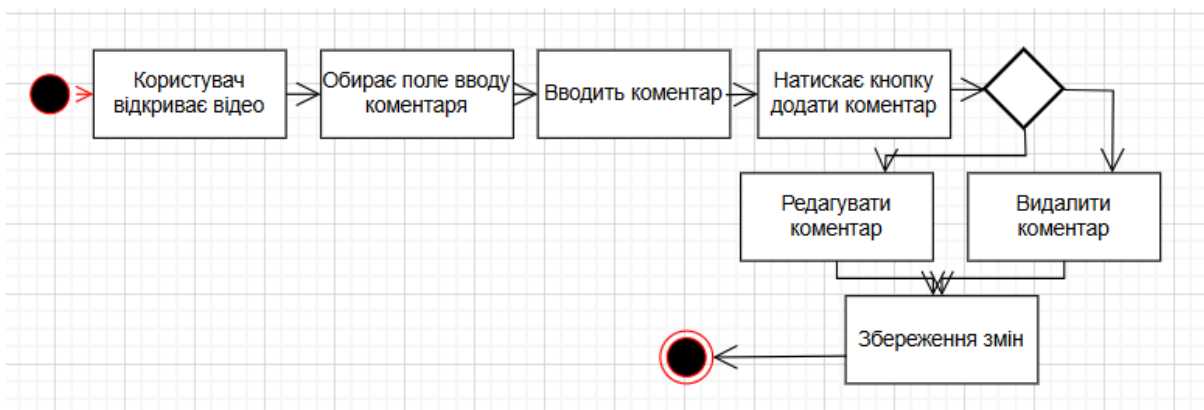


Рисунок 2.4 – UML-діаграма активності: створення коментаря користувачем

Для кращого розуміння логіки збереження плейлиста користувачем було змодельовано послідовність основних дій цього процесу. Користувач ініціює запит шляхом натискання відповідної кнопки, після чого платформа виконує звернення до бази даних з метою фіксації вибраного контенту у списку

збережених. Результат дій повертається у вигляді оновленого інтерфейсу та повідомлення про успішне виконання операції (рис.2.5).

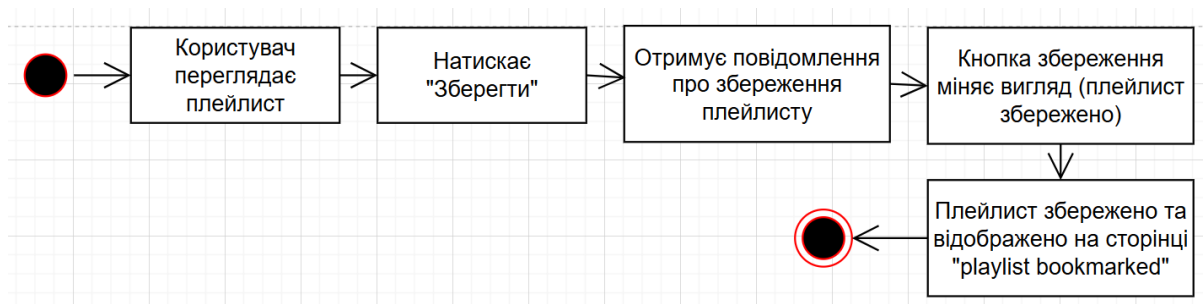


Рисунок 2.5 – UML-діаграма активності: збереження плейлиста користувачем

Для відображення логіки взаємодії користувача із відео-контентом було проаналізовано процес встановлення позначки «вподобано», що зображено на рисунку 2.6. Після натискання на відповідну іконку, платформа формує запит, який фіксує дію в базі даних. Далі відбувається оновлення інтерфейсу для відображення актуального стану – зміна іконки та оновлення кількості вподобань.

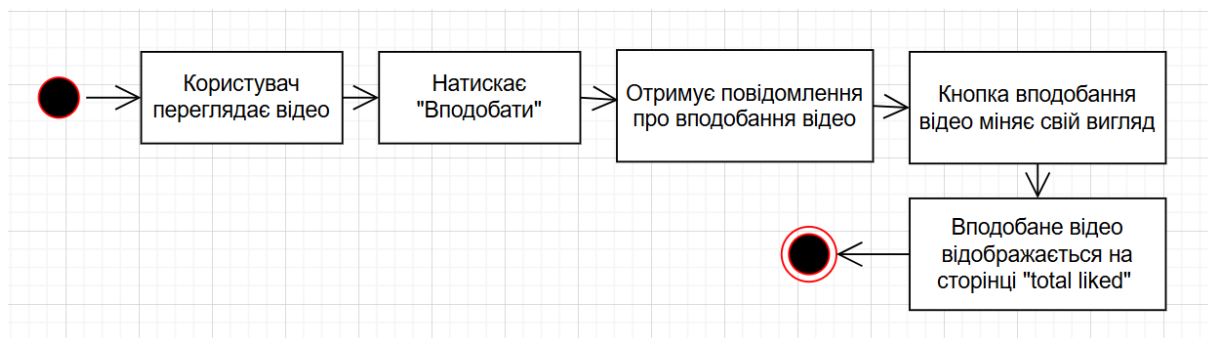


Рисунок 2.6 – UML-діаграма активності: вподобання відео користувачем

Для опису логіки взаємодії користувача з функціоналом пошуку курсів було змодельовано відповідний процес, представлений на рисунку 2.7. Після введення пошукового запиту та натискання кнопки «Пошук», платформа формує SQL-запит до бази даних з метою знаходження курсів, що відповідають заданим критеріям. У разі наявності збігів, результат виводиться на екран у вигляді

списку курсів із ключовою інформацією. Якщо відповідних записів не знайдено або запит був порожнім, користувач отримує повідомлення про відсутність результатів або необхідність введення пошуку.

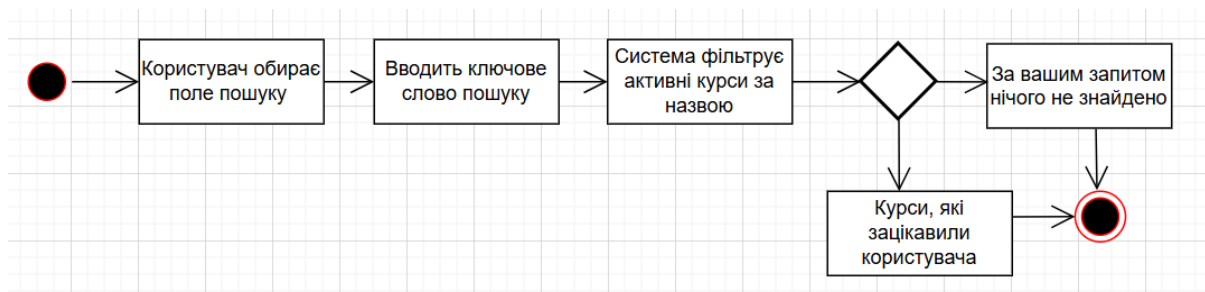


Рисунок 2.7 – UML-діаграма активності: пошук курсу користувачем

З метою структурування логіки зворотного зв'язку з користувачем було проаналізовано процес заповнення контактної форми, зображеного на рисунку 2.8. У разі успішного проходження перевірки формується запит до сервера, який зберігає повідомлення в базі даних. Після цього користувач отримує повідомлення про успішну відправку або відповідне повідомлення про помилку.

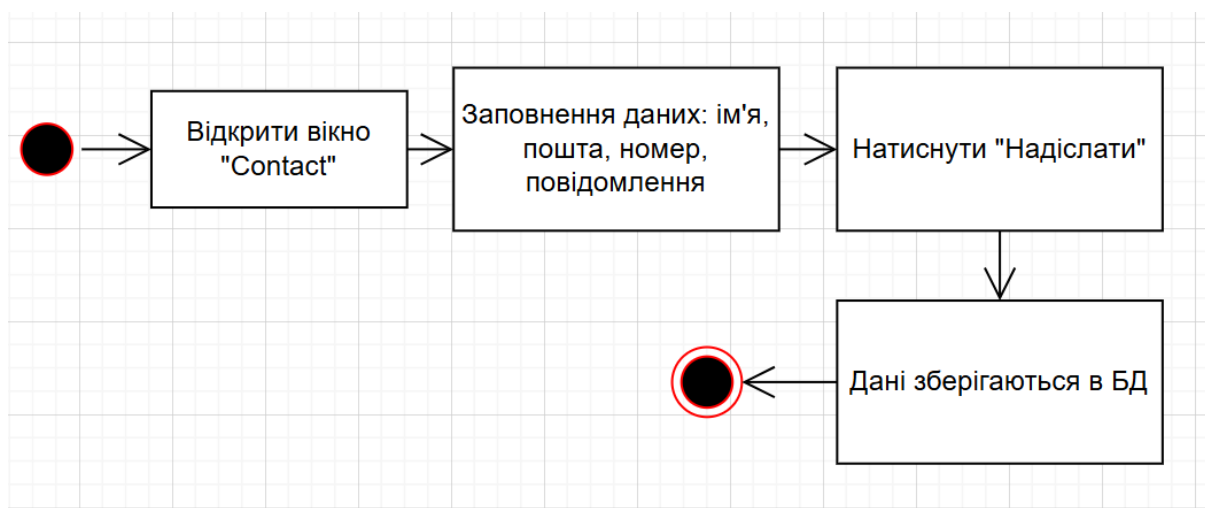


Рисунок 2.8 – UML-діаграма активності: заповнення форми get in touch користувачем

Після моделювання поведінки звичайного користувача важливо окремо розглянути роль адміністратора в системі. На відміну від користувача, який взаємодіє переважно з уже створеним контентом, адміністратор відповідає за повноцінне керування структурою та наповненням платформи, а також за контроль за активністю користувачів.

Саме тому наступним етапом стало моделювання ключових адміністративних процесів – від створення плейлистів до модерації коментарів – за допомогою діаграм активності, які дозволяють відобразити послідовність дій та логіку прийняття рішень у межах системи.

На рисунку 2.9 наведено діаграму активності, яка моделює взаємодію адміністратора з функціоналом перегляду коментарів та охоплює етапи від завантаження відповідного інтерфейсу до отримання та відображення коментарів з бази даних, а також передбачає дії адміністратора щодо подальшої обробки отриманої інформації.

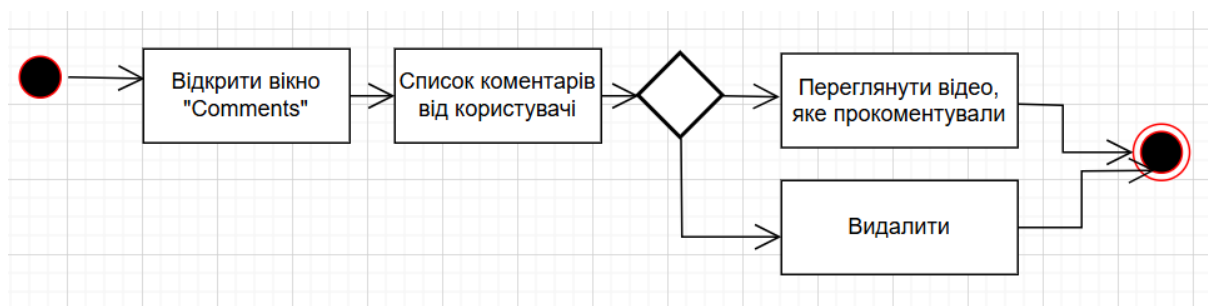


Рисунок 2.9 – UML-діаграма активності: перегляд коментарів адміністратором

У межах адміністративних сценаріїв наступним етапом стало моделювання процесу перегляду вподобаного контенту, створеного адміністратором (рис. 2.10). Дана діаграма відображає послідовність дій, що дозволяє адміністратору здійснити запит до платформи для отримання переліку матеріалів, які отримали позначку «вподобано» від користувачів. У результаті платформа фільтрує контент за відповідними критеріями та виводить актуальну інформацію з можливістю подальшого аналізу або взаємодії.

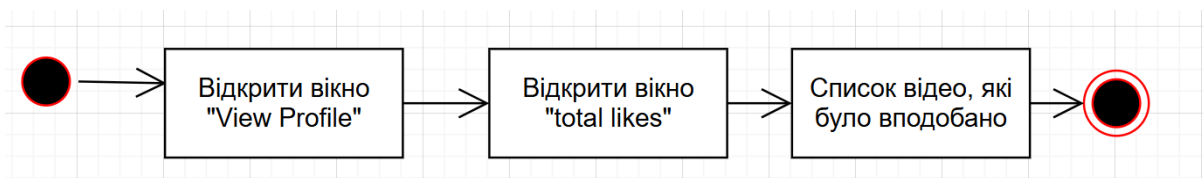


Рисунок 2.10 – UML-діаграма активності: перегляд вподобаного контенту, який створив адміністратор

У межах моделювання адміністративної частини системи було також проаналізовано функціонал пошуку, що дозволяє адміністратору швидко знаходити необхідний контент або записи у базі даних (рис. 2.11). На діаграмі зображено послідовність дій, від введення пошукового запиту до виведення релевантних результатів. У разі відсутності збігів платформа повідомляє про це через інтерфейс адміністратора.

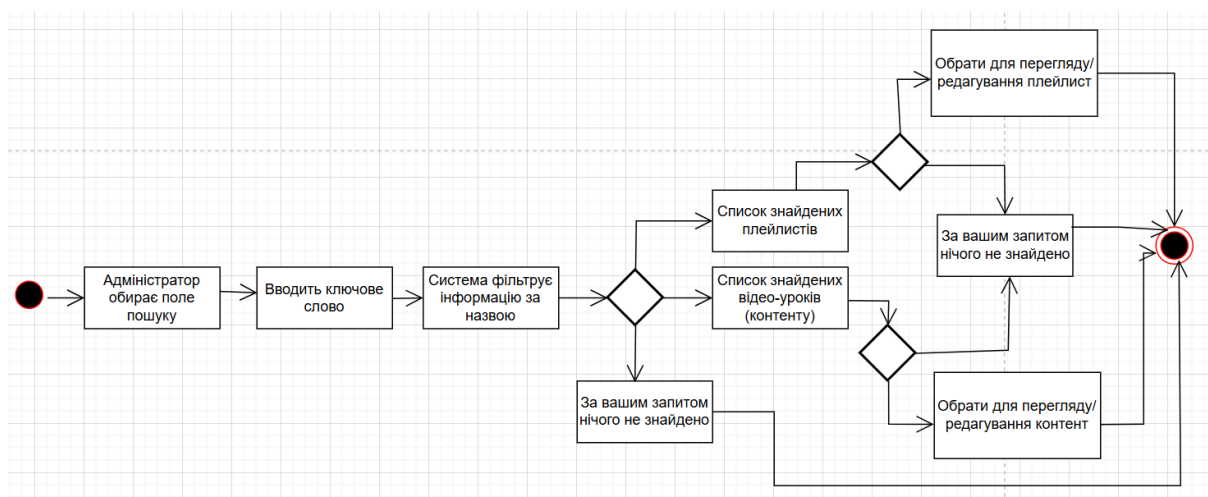


Рисунок 2.11 – UML-діаграма активності: функціонал пошуку адміністратором

Діаграма, представлена на рисунку 2.12, ілюструє адміністраторський сценарій створення навчального плейлиста з подальшим його наповненням навчальним контентом. Процес включає перевірку коректності введених даних і супроводжується відповідними повідомленнями про успішне виконання дій. Важливо зазначити, що при встановленні для плейлиста або доданих матеріалів статусу «неактивний», вони не відображаються звичайним користувачам в

інтерфейсі платформи. Це дозволяє адміністратору попередньо готувати або редагувати контент без його передчасної публікації.

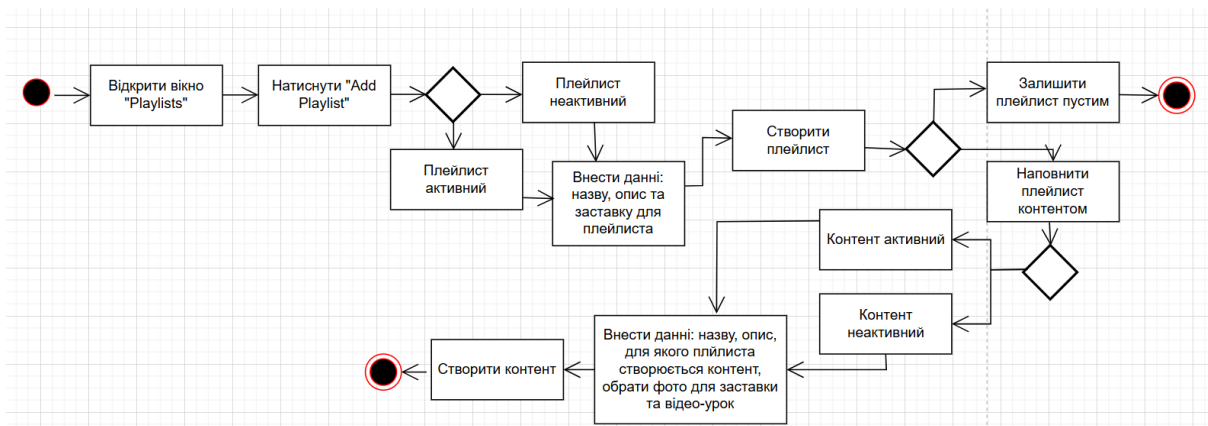


Рисунок 2.12 – UML-діаграма активності: створення та наповнення плейлиста адміністратором

На завершення моделювання адміністративних функцій було побудовано діаграму, яка відображає логіку перегляду профілю адміністратора (рис. 2.13).

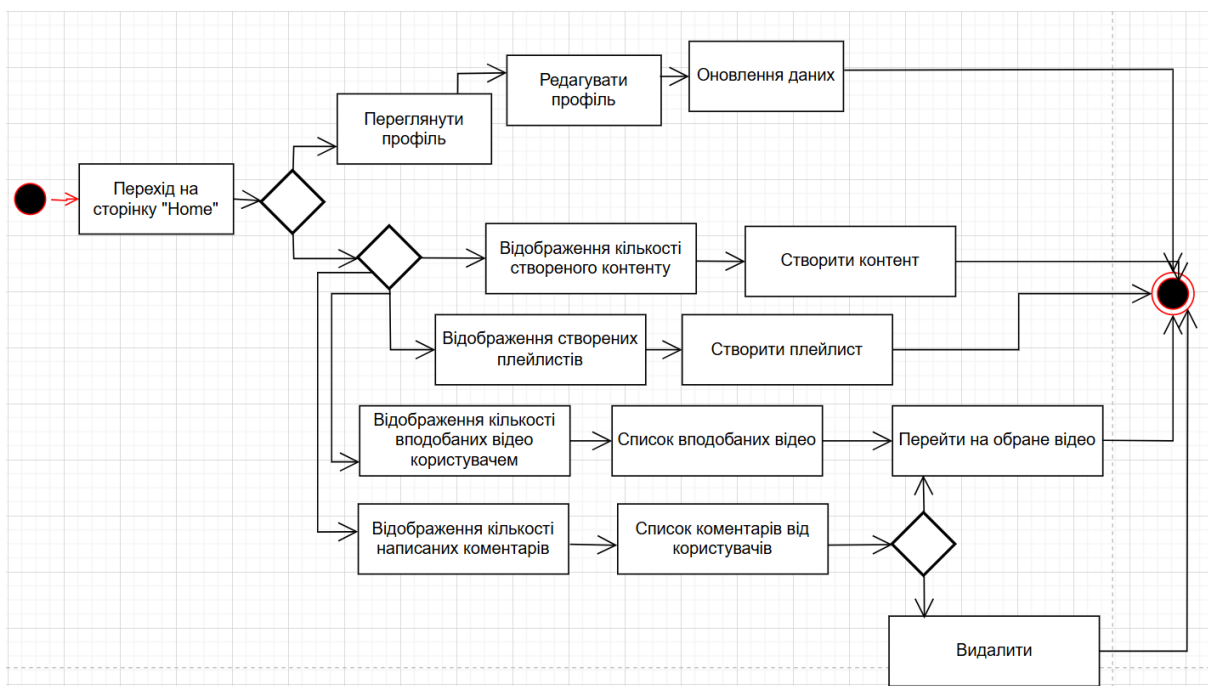


Рисунок 2.13 – UML-діаграма активності: перегляд профілю адміністратором

Після ініціювання відповідного запиту платформа звертається до бази даних для отримання актуальної інформації, яка потім відображається у структурованому та зручному для сприйняття інтерфейсі. Це забезпечує адміністратору швидкий доступ до ключових відомостей про власний обліковий запис, що, у свою чергу, сприяє ефективному керуванню користувачами та загальною координацією діяльності в межах платформи.

Таким чином, результати моделювання стануть фундаментом для подальших етапів розробки: проєктування бази даних, створення користувацького інтерфейсу, написання серверної логіки та реалізації механізмів взаємодії між користувачем і системою.

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

У процесі розробки онлайн-платформи було обрано набір вебтехнологій, що дозволило реалізацію проєкту. Серед них – мова розмітки HTML, мова стилів CSS, мова програмування JavaScript, серверна мова програмування PHP, система керування базами даних MySQL, а також середовище локального розгортання XAMPP. Саме поєднання цих технологій дозволило створити повноцінну багатофункціональну систему з чітким поділом на візуальну частину й серверну логіку.

Для розробки зовнішнього вигляду платформи було обрано стандартизовану мову розмітки, яка використовується для створення та структурування сторінок – HTML. Вона не є мовою програмування, оскільки не містить логіки виконання, проте саме вона виступає фундаментом для будь-якого вебресурсу. За допомогою HTML буде сформовано логічну ієрархію інтерфейсу користувача та адміністратора: створено каркас головної сторінки, особистого кабінету, реєстрації, сторінки курсів та інших розділів [10].

Після цього до роботи буде залучено CSS – спеціальну мову стилю сторінок, яка визначає зовнішній вигляд HTML-елементів та дозволяє задавати

кольори, розміри, шрифти, відступи, анімації, адаптивність та інші параметри, що впливають на зовнішнє сприйняття сайту. Стили накладаються один на одного відповідно до ієрархії, що дозволяє контролювати пріоритети оформлення. У сучасній розробці використовуються такі підходи, як Flexbox і Grid для побудови гнучких макетів, а також медіа-запити для реалізації адаптивного дизайну. Саме завдяки каскадним таблицям стилів було реалізовано оформлення сторінок, підбрано шрифти, кольори, відступи, а також створено адаптивний дизайн, що коректно відображається на різних типах пристроїв [11].

Після визначення зовнішнього вигляду було розпочато роботу над логікою взаємодії та динамічними можливостями платформи. Для цього я обрала JavaScript – мову програмування, яку використовують в основному для клієнтської логіки на вебсторінках. Вона дозволяє зробити сайт інтерактивним: реагувати на дії користувача, змінювати вміст без перезавантаження сторінки, перевіряти введені дані у формах, створювати динамічні елементи [12]. У сучасній розробці JS є надзвичайно потужним інструментом – з нього починаються популярні фреймворки й бібліотеки. У межах цього проєкту за допомогою JavaScript буде реалізована робота кнопок – їхня реакція на натискання, виведення повідомлень, керування модальними вікнами, валідація форм тощо. Його використання дозволить зробити інтерфейс живим, інтерактивним і комфортним для користувача.

Основу серверної частини складатиме мова PHP, яка дозволяє створювати динамічні вебсторінки та вебзастосунки. Її код виконується на сервері, генеруючи HTML, який потім передається користувачу. PHP тісно інтегрується з базами даних і особливо популярний для створення систем, де потрібна взаємодія з користувачем: обробка форм, авторизація, робота з файлами, відправлення електронної пошти, керування сесіями тощо [13].

Зберігання усіх даних, зокрема інформації про користувачів, курси, коментарі, збереження та вподобання, буде реалізовано за допомогою MySQL – реляційної системи керування базами даних. Вона дозволяє створювати таблиці, встановлювати зв'язки між ними, здійснювати пошук, фільтрацію, оновлення та

видалення записів. MySQL використовує мову SQL для взаємодії з даними. Вона ідеально підходить для створення багатокористувацьких систем, де необхідно зберігати інформацію про облікові записи, дії користувачів, повідомлення, файли тощо. Через використання PHP і MySQL буде забезпечено повноцінний двосторонній обмін даними між користувачем і системою [14].

Під час розробки проєкту використовуватиметься середовище XAMPP – безкоштовне програмне середовище для локальної розробки, яке включає в себе вебсервер Apache, базу даних MariaDB з підтримкою MySQL, а також інтерпретатори для PHP і Perl. Його головна перевага – простота встановлення та налаштування. XAMPP дозволяє розгортати вебсайти локально, тестувати код, створювати бази даних і бачити результат роботи до того, як продукт буде розміщений на реальному сервері. Завдяки XAMPP розробка платформи здійснюватиметься локально, що дасть можливість швидко тестувати зміни, працювати з базою даних у реальному часі та налагоджувати серверну логіку без необхідності звертання до зовнішнього хостингу [15].

Таким чином, створення платформи відбуватиметься поетапно – спочатку формуватиметься візуальна частина з використанням HTML і CSS, що дозволить побачити загальну структуру та вигляд інтерфейсу. Далі до проєкту додаватиметься динамічна поведінка через JavaScript і реалізовуватиметься повна функціональність на основі PHP та MySQL.

Висновки по розділу 2

В результаті системного моделювання було створено комплексну картину логіки онлайн-платформи, що охоплює взаємодію як адміністратора, так і пересічного користувача. Було створено кілька UML-діаграм, які ілюструють логіку роботи платформи на основі розглянутих вимог та функціональних сценаріїв.

З'ясування функціональних вимог як для адміністраторів, так і для студентів за допомогою моделювання полегшило структурування архітектури

майбутньої платформи та допомогло запобігти важливим помилкам ще до того, як вона була введена в експлуатацію. Однією з особливостей системи була здатність чітко визначати права доступу; інтерфейс динамічно адаптується до ролі користувача. У той час як адміністратор має розширені можливості для управління курсами, редагування даних, модерації та аналітики, звичайний користувач взаємодіє з контентом, переглядає курси та виконує завдання.

У межах подальшої реалізації платформи передбачено використання HTML, CSS і JavaScript для створення інтерактивного користувацького інтерфейсу, PHP – для реалізації серверної логіки обробки запитів, а також системи управління базами даних MySQL для зберігання та обробки інформації. Застосування цих мов програмування та технологій забезпечить стабільність, масштабованість і зручність користування створеним програмним продуктом.

РОЗДІЛ 3

РОЗРОБКА ОНЛАЙН-ПЛАТФОРМИ ДЛЯ ОСВІТНЬОГО ПРОЦЕСУ З ВИКОРИСТАННЯМ JAVASCRIPT ТА PHP

3.1 Практична реалізація об'єкта проєктування

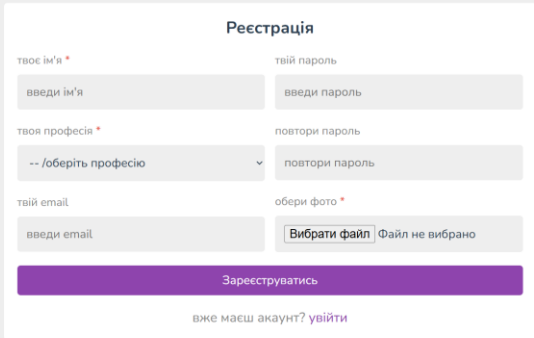
На етапі розробки було безпосередньо реалізовано функціональну частину вебплатформи згідно з вимогами та логікою, визначеними під час моделювання. Роботу поділено на дві основні частини: серверну (бекенд), реалізовану мовою програмування PHP, та клієнтську (фронтенд), реалізовану засобами HTML, CSS і JavaScript.

PHP використовується для динамічної генерації контенту, реалізації логіки, взаємодії з базами даних та обробки запитів користувачів. Ви можете створювати сторінки в реальному часі на основі запитів і системних умов, оскільки весь PHP код інтегрований в структуру HTML.

Для реалізації серверної частини вебплатформи, яка побудована на PHP та MySQL, я використала XAMPP – відомий програмний пакет, що складається з Apache, MySQL (MariaDB) та PHP. За допомогою цього середовища швидко та легко налаштувала сервер на локальному комп'ютері для тестування платформи. Процес розробки та тестування значно спрощується завдяки XAMPP, який дозволяє запускати сервер і базу даних без необхідності ручного налаштування.

Щоб забезпечити керування вебплатформою та контроль за її вмістом, було реалізовано функцію реєстрації адміністратора. Адміністратор матиме більш розширені можливості порівняно зі звичайними користувачами (студентами), зокрема доступ до перегляду, редагування та видалення матеріалів, користувачів і коментарів.

Реєстрація адміністратора передбачає заповнення спеціальної форми (рис. 3.1), яка обробляється на сервері засобами PHP.



Реєстрація

твоє ім'я *

твій пароль

твоя професія *

повтори пароль

твій email

обери фото * Файл не вибрано

[вже маєш акаунт? увійти](#)

Рисунок 3.1 – Сторінка реєстрації адміністратора

На рисунку 3.2 зображено вигляд сторінки «Плейлисти» від імені адміністратора, де демонструються доступні для керування елементи, зокрема можливість перегляду списку плейлистів, редагування їхнього вмісту, видалення, а також додавання нових плейлистів шляхом взаємодії з відповідними елементами.

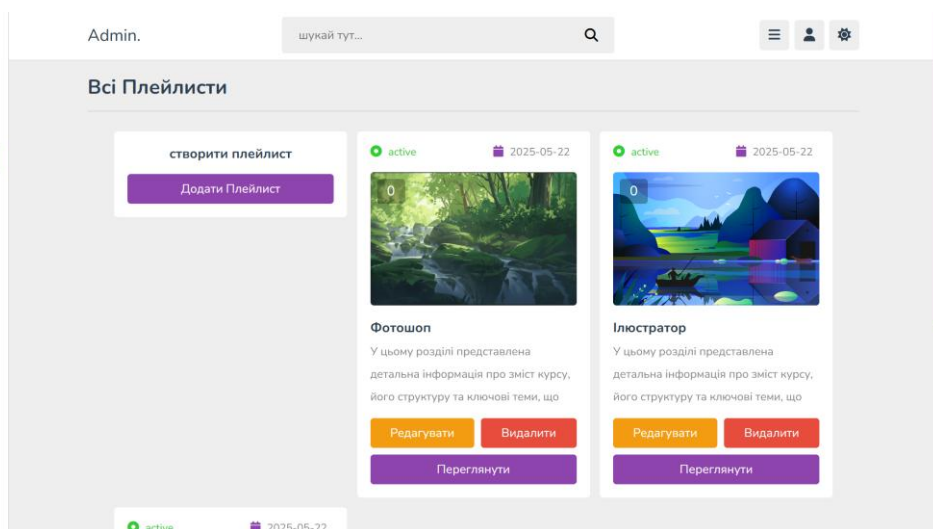


Рисунок 3.2 – Сторінка «Playlists» адміністратора

Дані, що вводяться, проходять перевірку, хешування пароля та збереження в базі даних у відповідній таблиці, після чого адміністратор авторизується у системі (рис. 3.3).

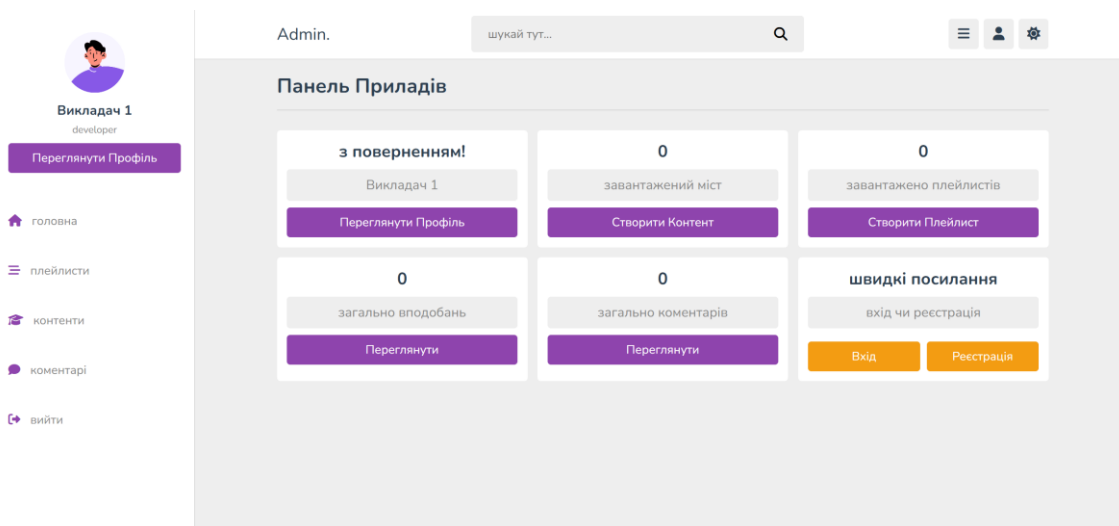


Рисунок 3.3 – Сторінка авторизованого адміністратора

У лістингу 3.1 наведено фрагменту коду, який демонструє реалізацію цієї функції.

Лістинг 3.1 – Реєстрація адміністратора

```
document.addEventListener('DOMContentLoaded', () => {
  let body = document.body;
  let profile = document.querySelector('.header .flex .profile');
  let searchForm = document.querySelector('.header .flex .search-form');
  let sideBar = document.querySelector('.side-bar');
  let toggleBtn = document.querySelector('#toggle-btn');
  let darkMode = localStorage.getItem('dark-mode');

  let userBtn = document.querySelector('#user-btn');
  if (userBtn) {
    userBtn.onclick = () => {
      profile.classList.toggle('active');
      searchForm.classList.remove('active');
    };
  }

  let searchBtn = document.querySelector('#search-btn');
  if (searchBtn) {
    searchBtn.onclick = () => {
      searchForm.classList.toggle('active');
      profile.classList.remove('active');
    };
  }

  let menuBtn = document.querySelector('#menu-btn');
```

```

if (menuBtn) {
  menuBtn.onclick = () => {
    sideBar.classList.toggle('active');
    body.classList.toggle('active');
    footer.classList.toggle('active');
  };
}

let closeBar = document.querySelector('#close-bar');
if (closeBar) {
  closeBar.onclick = () => {
    sideBar.classList.remove('active');
  };
}

window.onscroll = () => {
  profile.classList.remove('active');
  searchForm.classList.remove('active');

  if (window.innerWidth < 1200) {
    sideBar.classList.remove('active');
    body.classList.remove('active');
  }
};

```

кінець лістингу 3.1

Для забезпечення доступу до навчального контенту користувачам необхідно створити особистий обліковий запис. Реєстрація користувача включає заповнення форми з основною інформацією: ім'я, електронна пошта, пароль і фотографія профілю (рис.3.4).

The screenshot shows a web application interface for user registration. On the left, there is a sidebar with a login/register section and a menu. The main content area displays a registration form titled "Зареєструватися". The form includes fields for name, email, password, and profile picture, with a "Зареєструватися" button at the bottom.

Рисунок 3.4 – Сторінка реєстрації користувача

Уся введена інформація перевіряється на стороні сервера за допомогою PHP, після успішної перевірки дані зберігаються в базі даних, а користувач авторизується в системі (рис. 3.5).

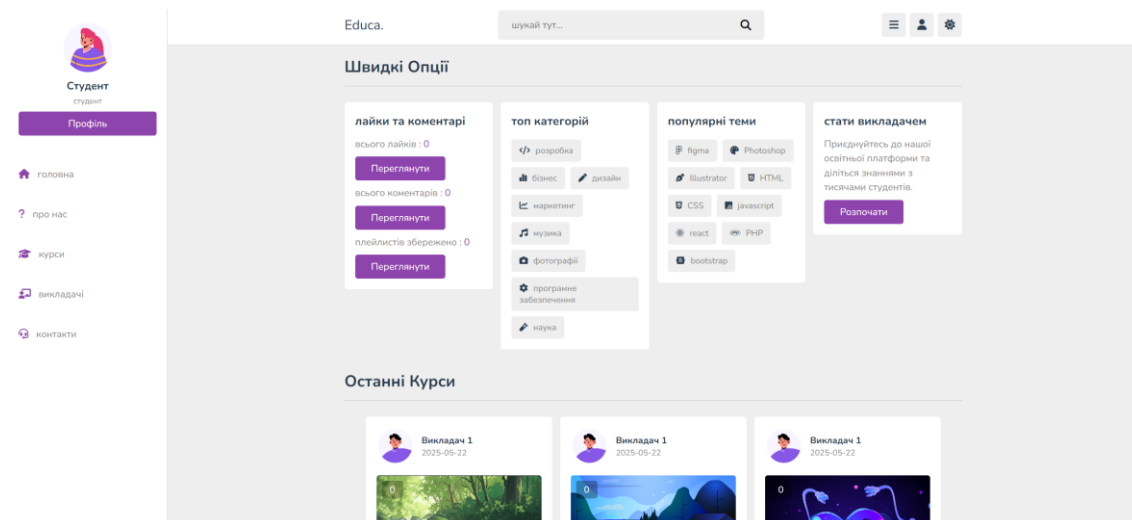


Рисунок 3.5 – Сторінка авторизації користувача

У лістингу 3.2 наведено фрагмент коду, який демонструє реалізацію авторизації користувача після реєстрації.

Лістинг 3.2 – Реєстрація користувача

```
document.addEventListener('DOMContentLoaded', () => {

    let body = document.body;
    let profile = document.querySelector('.header .flex .profile');
    let searchForm = document.querySelector('.header .flex .search-form');
    let sideBar = document.querySelector('.side-bar');
    let toggleBtn = document.querySelector('#toggle-btn');
    let darkMode = localStorage.getItem('dark-mode');

    let userBtn = document.querySelector('#user-btn');

    if (userBtn) {
        userBtn.onclick = () => {
            profile.classList.toggle('active');
            searchForm.classList.remove('active');
        };
    }

    let searchBtn = document.querySelector('#search-btn');
```

```

if (searchBtn) {
  searchBtn.onclick = () => {
    searchForm.classList.toggle('active');
    profile.classList.remove('active');
  };
}

let menuBtn = document.querySelector('#menu-btn');

if (menuBtn) {
  menuBtn.onclick = () => {
    sideBar.classList.toggle('active');
    body.classList.toggle('active');
    footer.classList.toggle('active');
  };
}

let closeBar = document.querySelector('#close-bar');

if (closeBar) {
  closeBar.onclick = () => {
    sideBar.classList.remove('active');
  };
}

window.onscroll = () => {
  profile.classList.remove('active');
  searchForm.classList.remove('active');

  if (window.innerWidth < 1200) {
    sideBar.classList.remove('active');
    body.classList.remove('active');
  }
};

```

кінець лістингу 3.2

У процесі реалізації функціоналу вебплатформи було впроваджено зручну та інтуїтивно зрозумілу систему пошуку навчальних курсів за ключовими словами, яка дозволяє користувачам швидко знаходити необхідний контент серед великої кількості матеріалів. Такий підхід суттєво підвищує ефективність взаємодії з інтерфейсом, сприяє економії часу та забезпечує легкий доступ до релевантного навчального контенту відповідно до потреб користувача (рис. 3.6).

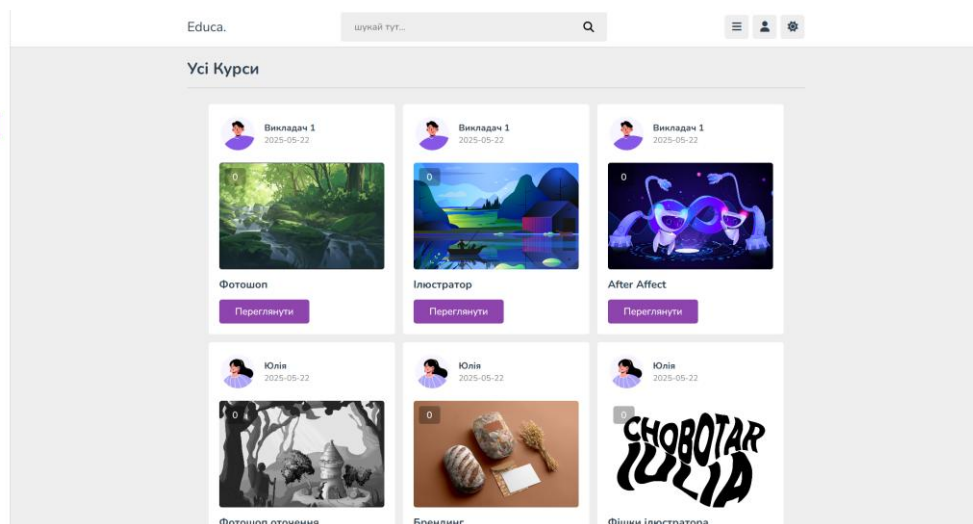


Рисунок 3.6 – Сторінка наявних курсів

Завдяки цій функції користувач має можливість оперативно знаходити потрібний навчальний матеріал без необхідності переглядати весь перелік доступних курсів. Пошуковий механізм фільтрує контент відповідно до введеної назви або її частини, що дозволяє максимально точно відображати лише релевантні результати (рис. 3.7).

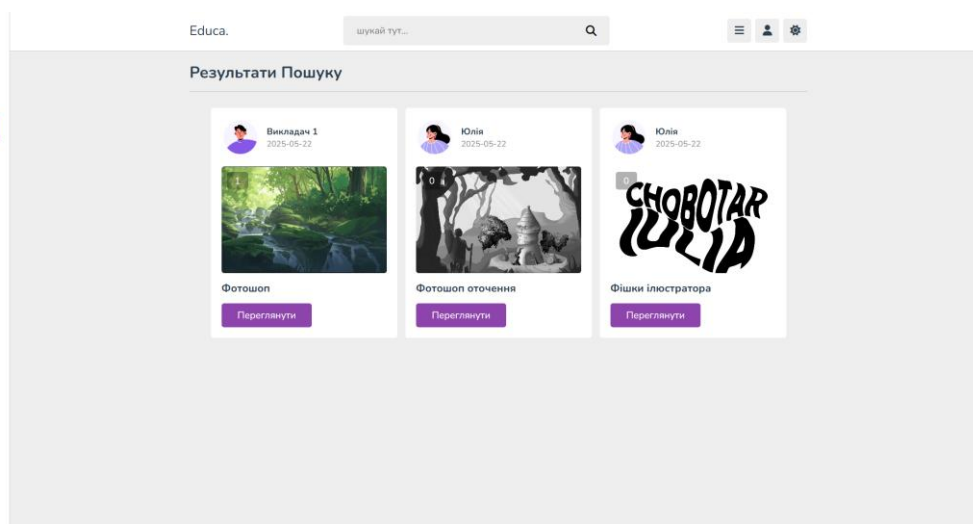


Рисунок 3.7 – Пошук курсів

З технічного боку реалізація функції передбачає обробку введеного значення у формі пошуку, створення SQL-запиту та виведення відповідних результатів. При цьому враховано обробку випадків, коли користувач не вводить

жодного значення або коли запит не дає результату – у таких ситуаціях виводиться відповідне повідомлення (ліст. 3.3).

Лістинг 3.3 – Реалізація пошуку курсу

```
<?php
    if(isset($_POST['search_box']) or isset($_POST['search_btn'])){
        $search_box = $_POST['search_box'];

        $select_courses = $conn->prepare("SELECT * FROM `playlist` WHERE
title LIKE '%{$search_box}%' AND status = ? ORDER BY date DESC");
        $select_courses->execute(['active']);
        if($select_courses->rowCount() > 0){
            while($fetch_course = $select_courses-
>fetch(PDO::FETCH_ASSOC)){
                $course_id = $fetch_course['id'];

                $count_course = $conn->prepare("SELECT * FROM `content`
WHERE playlist_id = ?");
                $count_course->execute([$course_id]);
                $total_courses = $count_course->rowCount();

                $select_tutor = $conn->prepare("SELECT * FROM `tutor`
WHERE id = ?");
                $select_tutor->execute([$fetch_course['tutor_id']]);
                $fetch_tutor = $select_tutor->fetch(PDO::FETCH_ASSOC);

            ?>
```

кінець лістингу 3.3

Якщо натиснута кнопка пошуку або заповнене поле пошуку, виконується SQL-запит до таблиці `playlist`, де знаходяться курси зі статусом «active», і заголовок яких відповідає пошуковому запиту. Для кожного знайденого курсу підраховується кількість пов'язаних матеріалів у таблиці `content`, а також отримується інформація про викладача з таблиці `tutor`.

Користувач також має змогу зберігати плейлисти у власному профілі, що дозволяє формувати добірку актуальних курсів. На рисунку 3.8 представлено інтерфейс до моменту збереження плейлиста, коли користувач ще не активував відповідну функцію.

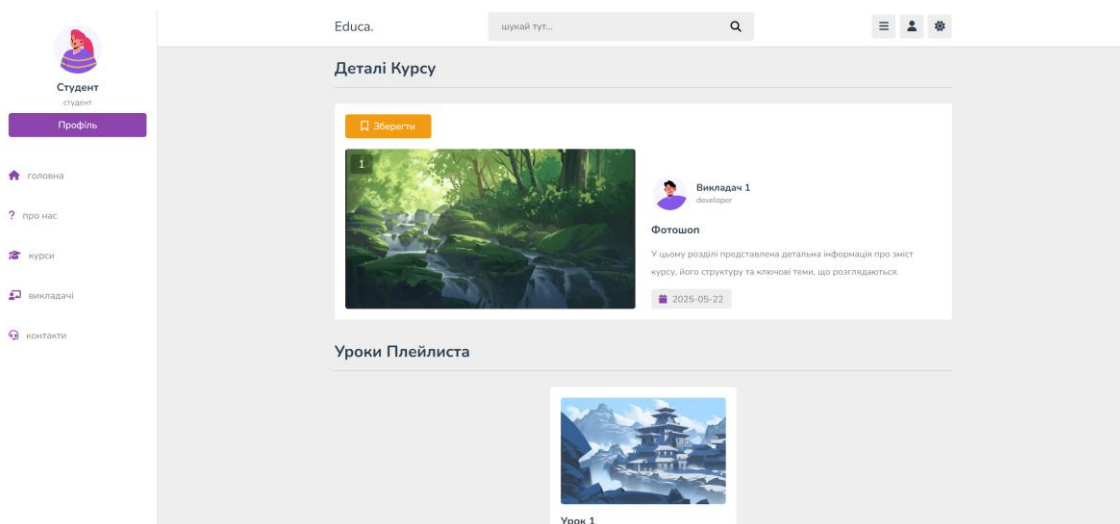


Рисунок 3.8 – Відображення курсу

Після збереження плейлиста платформа відображає оновлений стан із відповідним візуальним індикатором, як показано на рисунку 3.9.

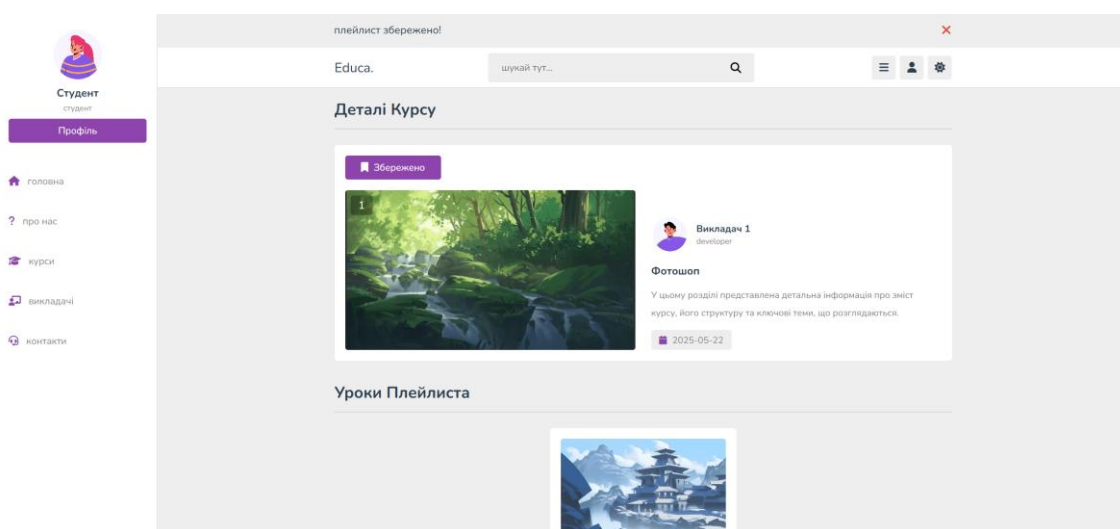


Рисунок 3.9 – Збереження курсу

Збережені матеріали стають доступними у спеціальному розділі профілю користувача, як зображено на рисунку 3.10.

Реалізація цієї функції передбачає створення відповідного запису в базі даних, що прив'язаний до ID користувача та ID збереженого плейлиста.

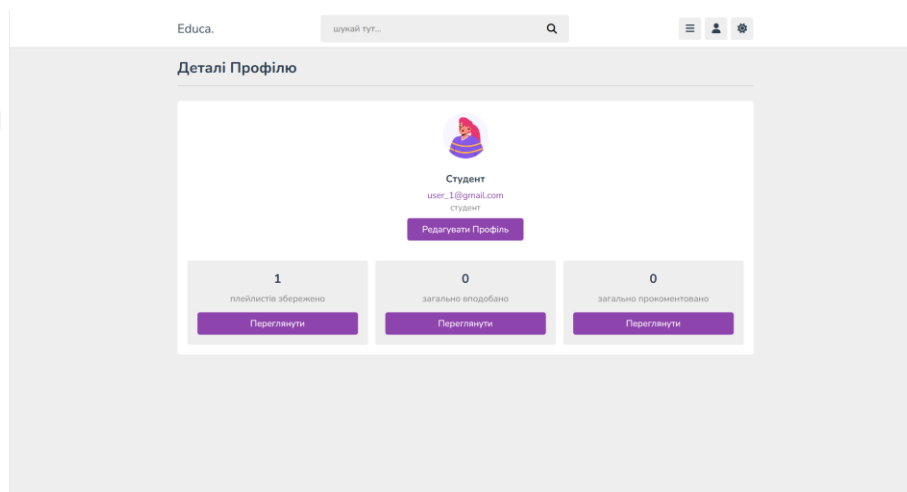


Рисунок 3.10 – Перегляд профілю

Аналогічна логіка застосована й до функції «вподобання», яка фіксує взаємодію користувача з відео та дозволяє формувати персоналізовану аналітику, реалізацію збереження плейлиста можна побачити на лістингу 3.4.

Лістинг 3.4 – Збереження плейлиста

```

if(isset($_POST['save_list'])){

    if($user_id != ''){
        $list_id = $_POST['list_id'];
        $list_id = filter_var($list_id, FILTER_SANITIZE_STRING);
        $verify_list = $conn->prepare("SELECT * FROM `bookmark` WHERE
user_id = ? AND playlist_id = ?");
        $verify_list->execute([$user_id, $list_id]);
        if($verify_list->rowCount() > 0){
            $remove_list = $conn->prepare("DELETE FROM `bookmark` WHERE
user_id = ? AND playlist_id = ?");
            $remove_list->execute([$user_id, $list_id]);
            $message[] = 'playlist removed!';
        }else{
            $add_list = $conn->prepare("INSERT INTO `bookmark` (user_id,
playlist_id) VALUES (?, ?)");
            $add_list->execute([$user_id, $list_id]);
            $message[] = 'playlist saved!';
        }
    }else{
        $message[] = 'please login first!';
    }
}

```

кінець лістингу 3.4

Функція коментування реалізована із захистом від порожніх повідомлень, це можна побачити на рисунку 3.11, що забезпечує коректність введених даних та запобігає збереженню пустих або випадкових коментарів, та у разі необхідності, дозволяє адміністратору вручну видаляти небажані або недоречні записи через відповідний інтерфейс керування з його ж сторони.

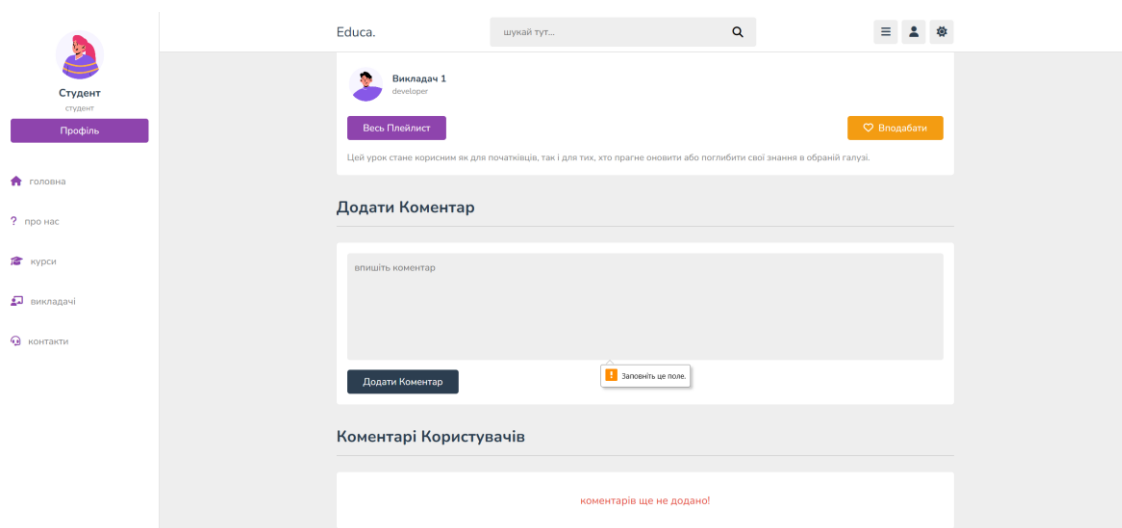


Рисунок 3.11 – Пустий коментар

На рисунку 3.12 показано, як відображається залишений користувачем коментар під певним відео, це дає змогу бачити реакції користувачів безпосередньо на сторінці перегляду.

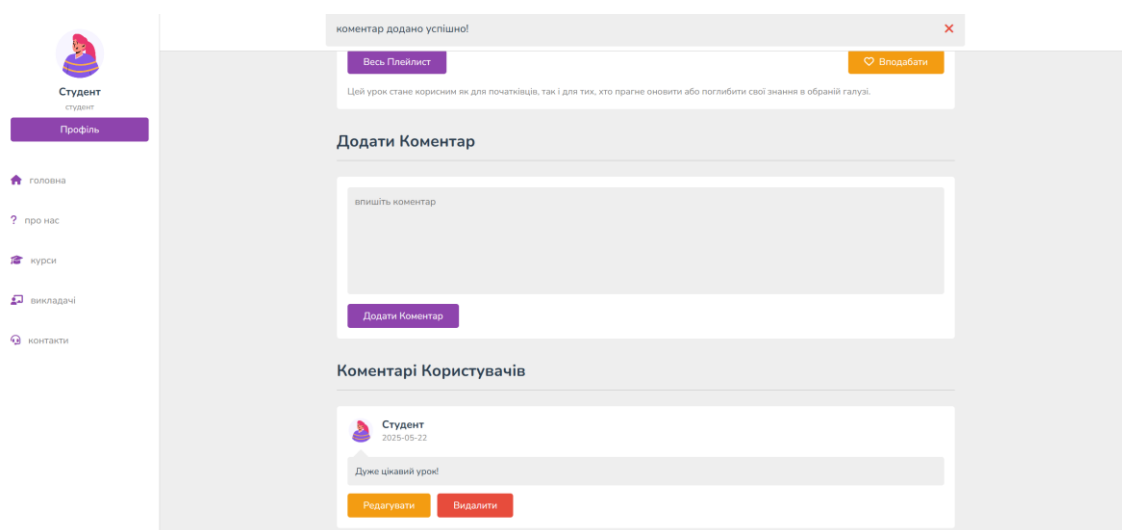


Рисунок 3.12 – Залишений коментар користувачем

Інтерфейс адміністратора, продемонстрований на рисунку 3.13, який має можливість переглядати всі коментарі та видаляти ті, що порушують етичні норми поведінки в інтернеті й поважного ставлення до інших користувачів.

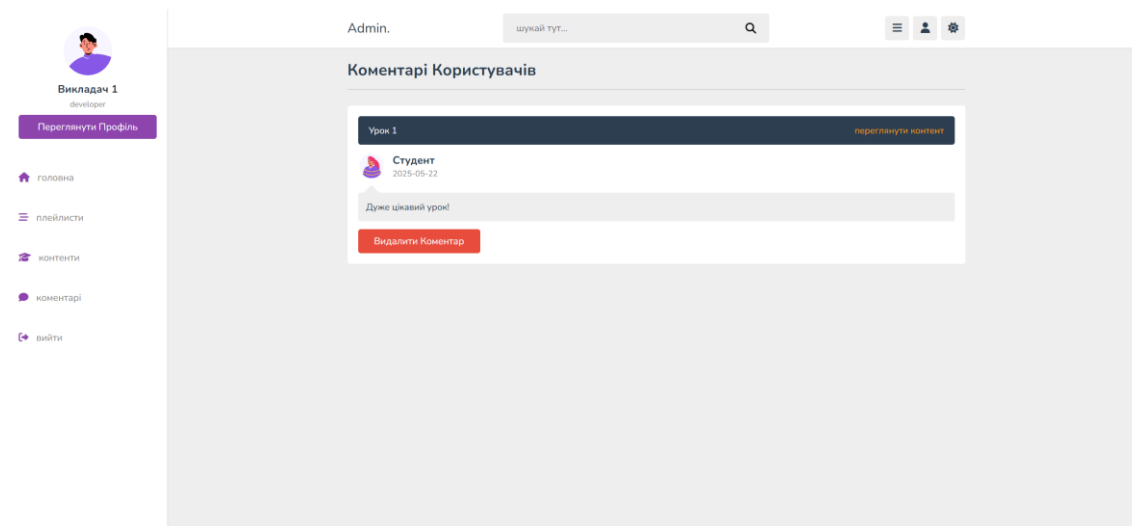


Рисунок 3.13 – Залишений коментар користувачем зі сторони адміністратора

Коментарі зберігаються у відповідній таблиці бази даних, що дозволяє зручно отримувати доступ до коментарів, забезпечуючи їх коректне відображення на сайті. Деталі структури цієї таблиці, яка містить необхідні зв'язки між коментарями та відео-уроками, наведено в лістингу 3.5.

Лістинг 3.5 – Видалення коментаря

```

if(isset($_POST['delete_comment'])){
    $delete_id = $_POST['comment_id'];
    $delete_id = filter_var($delete_id, FILTER_SANITIZE_STRING);
    $verify_comment = $conn->prepare("SELECT * FROM `comments` WHERE id =
?");
    $verify_comment->execute([$delete_id]);
    if($verify_comment->rowCount() > 0){
        $delete_comment = $conn->prepare("DELETE FROM `comments` WHERE id
= ? LIMIT 1");
        $delete_comment->execute([$delete_id]);
        $message[] = 'comment already successfully!';
    }else{
        $message[] = 'comment already deleted!';
    }
}

```

кінець лістингу 3.5

У лістингу 3.6 показано, як саме реалізується взаємозв'язок між коментарями та відповідними відео-уроками на рівні бази даних, що дозволяє ефективно управляти коментарями та забезпечувати їх правильну асоціацію з контентом.

Лістинг 3.6 – Редагування коментаря

```

if(isset($_POST['edit_comment'])){

    $edit_id = $_POST['edit_id'];
    $edit_id = filter_var($edit_id, FILTER_SANITIZE_STRING);

    $comment_box = $_POST['comment_box'];
    $comment_box = filter_var($comment_box, FILTER_SANITIZE_STRING);

    $verify_edit_comment = $conn->prepare("SELECT * FROM `comments` WHERE
id = ? AND comment = ? ");
    $verify_edit_comment->execute([$edit_id, $comment_box]);

    if($verify_edit_comment->rowCount() > 0){
        $message[] = 'comment already added!';
    }else{
        $update_comment = $conn->prepare("UPDATE `comments` SET comment =
? WHERE id = ? LIMIT 1");
        $update_comment->execute([$comment_box, $edit_id]);
        $message[] = 'comment updated successfully!';
    }
}

```

кінець лістингу 3.6

Для реалізації важливого функціоналу вебплатформи активно використовується JavaScript, який забезпечує інтерактивність і динамічні зміни контенту без необхідності перезавантаження сторінки. Він обробляє взаємодії з елементами інтерфейсу, такими як меню, профіль користувача і форма пошуку.

Зокрема, при натисканні на кнопки відкриваються або закриваються відповідні частини інтерфейсу, фрагмент коду якого лістингу 3.7.

Лістинг 3.7 – Обробка елків та кнопок

```

document.addEventListener('DOMContentLoaded', () => {
    let body = document.body;
    let profile = document.querySelector('.header .flex .profile');

```

```

    let searchForm = document.querySelector('.header .flex .search-
form');
    let sideBar = document.querySelector('.side-bar');
    let toggleBtn = document.querySelector('#toggle-btn');
    let darkMode = localStorage.getItem('dark-mode');

    let userBtn = document.querySelector('#user-btn');
    if (userBtn) {
        userBtn.onclick = () => {
            profile.classList.toggle('active');
            searchForm.classList.remove('active');
        };
    }

    let searchBtn = document.querySelector('#search-btn');
    if (searchBtn) {
        searchBtn.onclick = () => {
            searchForm.classList.toggle('active');
            profile.classList.remove('active');
        };
    }

    let menuBtn = document.querySelector('#menu-btn');
    if (menuBtn) {
        menuBtn.onclick = () => {
            sideBar.classList.toggle('active');
            body.classList.toggle('active');
            footer.classList.toggle('active');
        };
    }

    let closeBar = document.querySelector('#close-bar');
    if (closeBar) {
        closeBar.onclick = () => {
            sideBar.classList.remove('active');
        };
    }

    window.onscroll = () => {
        profile.classList.remove('active');
        searchForm.classList.remove('active');

        if (window.innerWidth < 1200) {
            sideBar.classList.remove('active');
            body.classList.remove('active');
        }
    };

```

3.2 Тестування та налагодження інформаційно-комп'ютерної системи

Після реалізації основної функціональності було проведено всебічне тестування вебплатформи. Основним підходом стало ручне тестування, у межах якого перевірялися всі ключові сценарії взаємодії – реєстрація та авторизація, створення та перегляд курсів, навігація по розділах, коректність заповнення форм і обробка некоректних або неповних даних. Ручне тестування дозволило виявити логічні помилки, проблеми з версткою та некоректну обробку окремих запитів.

Паралельно застосовувалися елементи модульного тестування – ізольована перевірка функцій або частин логіки, особливо в частині PHP-скриптів для взаємодії з базою даних. Це дозволило пересвідчитись у правильності виконання запитів, обробки помилок, валідації даних і захисту від типових атак (наприклад, SQL-ін'єкцій).

З метою покращення ефективності перевірки використовувалися засоби штучного інтелекту (ШІ). За допомогою LLM-моделей проводився аналіз коду на предмет потенційних помилок, неефективностей, вразливостей у логіці перевірки та обробки введення. Також ШІ допомагав формулювати типові сценарії тестування, генерувати приклади вхідних даних і визначати ризиковані зони коду.

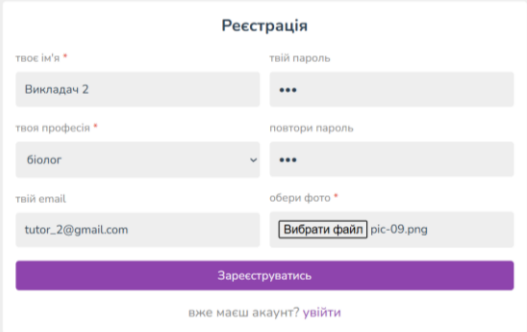
Окрім того, були застосовані елементи інтеграційного тестування, під час якого перевірялась коректна взаємодія між різними компонентами системи: клієнтською частиною, сервером і базою даних. Таке тестування дозволило виявити проблеми, що виникали при передачі даних, або неконсистентність між структурою бази й логікою в коді.

Задля підвищення зручності користування проводилося тестування інтерфейсу користувача (UI/UX-тестування). Було оцінено логіку розташування елементів, інтуїтивність навігації, доступність функцій та поведінку системи під час взаємодії з нею. Результати таких перевірок дали змогу оптимізувати

інтерфейс, зменшити кількість кроків для виконання типових задач і покращити загальний досвід користувача.

Усі знайдені недоліки фіксувалися, аналізувалися та усувалися на етапі розробки.

Першою проблемою в мене виникла реалізація функції реєстрації, що після реєстрації користувача платформа перенаправляла його на сторінку авторизації знову, хоча передбачалося перенаправлення на головну сторінку (рис. 3.14).

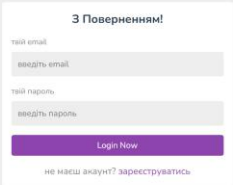


The screenshot shows a registration form titled "Регістрація". It contains the following fields and elements:

- твоє ім'я ***: Text input with "Викладач 2".
- твій пароль**: Password input with masked characters "•••".
- твоя професія ***: Dropdown menu with "біолог" selected.
- повтори пароль**: Password input with masked characters "•••".
- твій email**: Text input with "tutor_2@gmail.com".
- обери фото ***: File selection button labeled "Вибрати файл" and a filename "pic-09.png".
- Зареєструватись**: A prominent purple button.
- вже маєш акаунт? увійти**: A link at the bottom.

Рисунок 3.14 – Заповнення форми адміністратором

Однак, замість головної сторінки, платформа направляла користувача на сторінку авторизації, що відображено на рис. 3.15.



The screenshot shows a login form titled "З поверненням!". It contains the following fields and elements:

- твій email**: Text input with placeholder "введи email".
- твій пароль**: Password input with placeholder "введи пароль".
- Login Now**: A purple button.
- не маєш акаунт? зареєструватись**: A link at the bottom.

Рисунок 3.15 – Невірне перенаправлення після реєстрації адміністратора

Причина цього полягала в неправильному використанні cookie для збереження даних сесії.

Проблему було усунено шляхом заміни cookie на сесійні змінні, що дозволяє коректно зберігати статус користувача після реєстрації і здійснювати перенаправлення на головну сторінку (рис. 3.16).

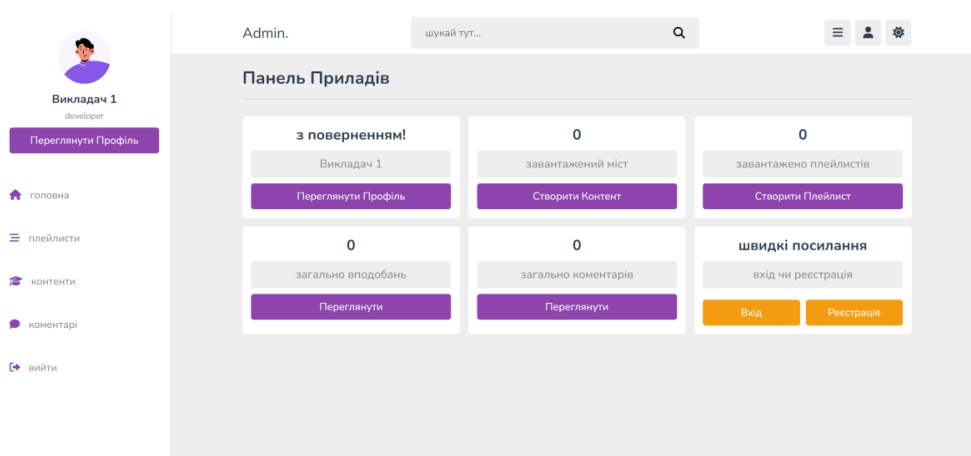


Рисунок 3.16 – Вірне перенаправлення після реєстрації адміністратора

Наступна помилка в мене також з'явилась зі сторони адміністратора – при спробі змінити фото для плейлиста, також змінювалось фото профілю, і навпаки (рис. 3.17).

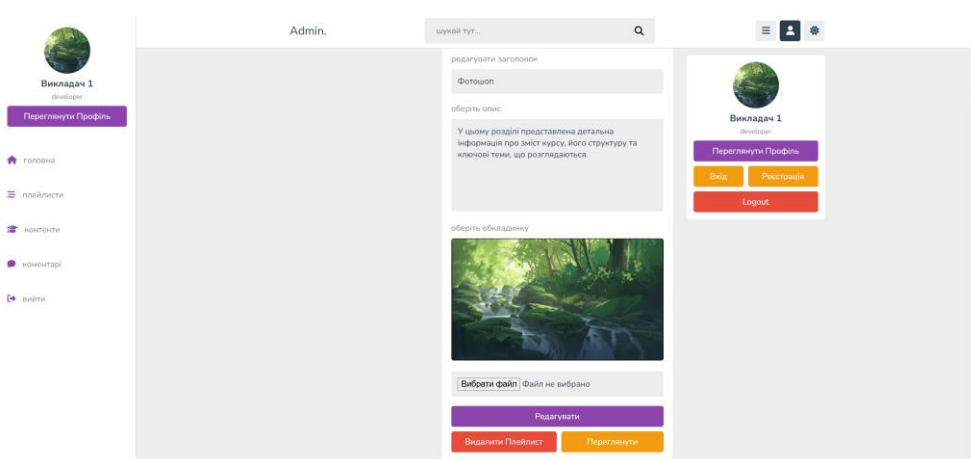


Рисунок 3.17 – Зміна зайвих фото при редагуванні

Це виникало через те, що для збереження зображень не було достатньо чіткої ідентифікації, яке фото належить до якого елемента, унаслідок чого виникали конфлікти при їх збереженні.

Для вирішення цієї проблеми було запроваджено використання унікальних префіксів до імен файлів, щоб уникнути конфліктів при збереженні зображень. Для імен файлів що належать до профілю було додано префікс `profile_` перед іменем файлу, що дозволило правильно зберігати зображення для самого профілю та плейлистів окремо.

Також, під час тестування було помічено, що при створенні першого контенту для плейлиста, усе працює коректно (рис. 3.18).

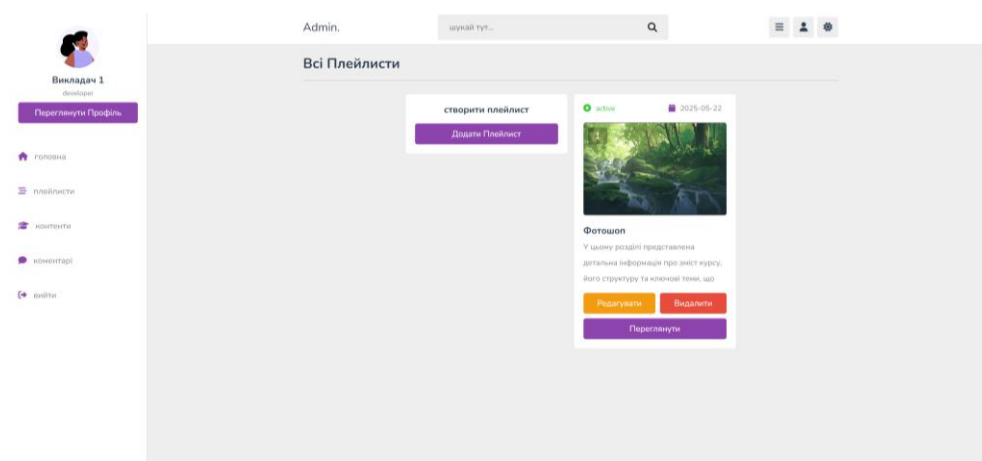


Рисунок 3.18 – Створення першого контенту

Однак при створенні наступних відео-уроків, фото з першого контенту змінювалось на те ж, що у новоствореного контенту.

Причиною цієї помилки була відсутність унікальних імен для файлів зображень, які прикріплюються до контенту. При завантаженні нового зображення з тим самим іменем, попередній файл у директорії сервера перезаписувався новим, оскільки система сприймала їх як один і той самий файл. У результаті зображення автоматично оновлювались у всіх записах, де воно використовувалося (рис. 3.19).

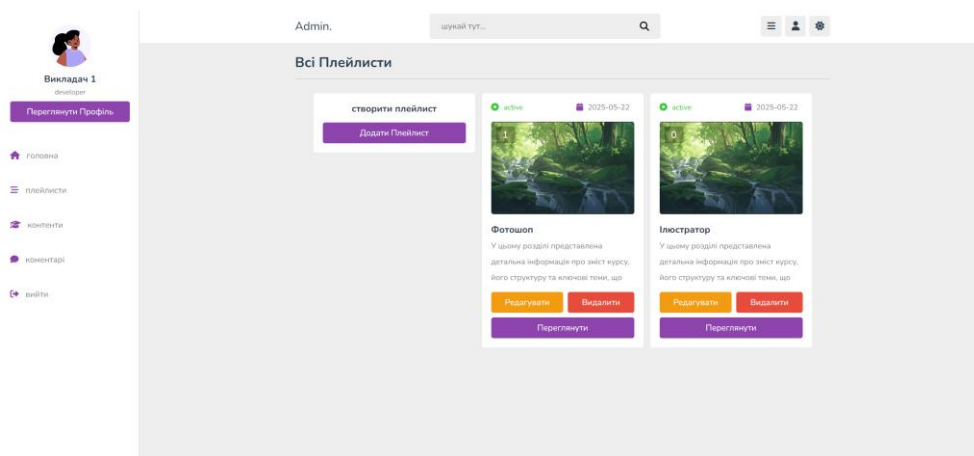


Рисунок 3.19 – Створення другого контенту

Проблему було усунено шляхом додавання префіксів до імен файлів, подібно до попереднього рішення. Це дозволило уникнути перезаписування зображень для різних елементів і забезпечило правильне збереження фото для кожного контенту окремо.

3.3 Проектування структури бази даних

Після завершення етапу логічного проектування бази даних, побудови діаграми сутність-зв'язок та визначення всіх необхідних таблиць і зв'язків між ними, які були зображені в розділі 2.1, здійснено реалізацію цієї структури у вигляді фізичної бази даних. Для цього використовувався інструмент phpMyAdmin, який забезпечує зручний графічний інтерфейс для взаємодії з системою управління базами даних MySQL. Робота в phpMyAdmin значно спрощує процес створення таблиць, визначення типів даних, встановлення ключів та інших атрибутів без потреби писати всі SQL-команди вручну.

У межах даного проєкту phpMyAdmin використовувався через локальний сервер XAMPP, що дозволяло тестувати функціональність бази даних без підключення до зовнішнього хостингу. Основна мета створення бази даних полягала в організації зберігання та обробки інформації, необхідної для роботи навчальної онлайн-платформи. Зокрема, структура охоплює таблиці для користувачів, викладачів, освітнього відеоконтенту, плейлистів, коментарів,

лайків, закладок та повідомлень, які надсилають відвідувачі через контактну форму. Усі ці таблиці створювалися відповідно до попередньо спроектованої структури, і між ними було встановлено відповідні зв'язки, щоб забезпечити цілісність та логічну взаємодію даних.

Таблиця `users` зберігає інформацію про звичайних користувачів платформи, включаючи їх ім'я, email, пароль та зображення профілю (рис. 3.20). Вона має зв'язок із таблицею `comments` за допомогою поля `user_id`, що дозволяє визначити, який користувач залишив певний коментар. Аналогічно, зв'язок із таблицею `likes` також базується на полі `user_id`, що дозволяє ідентифікувати користувача, який поставив лайк. Крім того, таблиця `users` пов'язана з таблицею `bookmark`, яка дозволяє користувачам зберігати улюблені плейлисти для швидкого доступу в майбутньому.

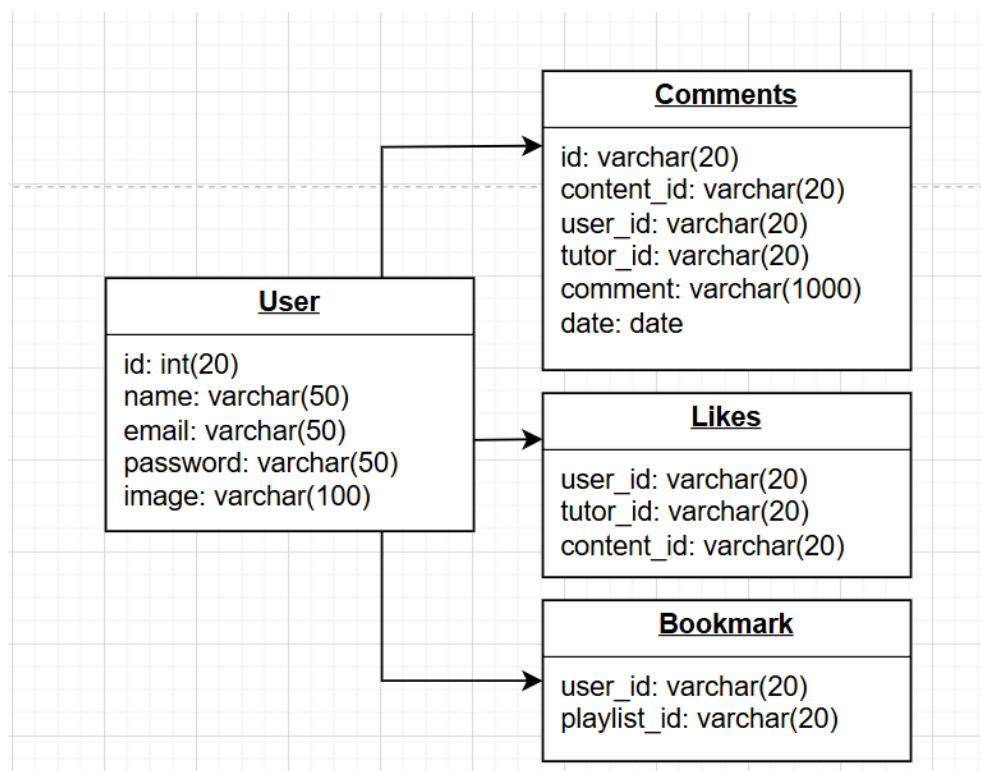


Рисунок 3.20 – Таблиця зв'язків User

Таблиця `tutor` містить інформацію про викладачів платформи – їх ім'я, професію, email, пароль та аватар (рис. 3.21). Вона має зв'язок із таблицею `playlist` через поле `tutor_id`, що дозволяє відстежувати, який викладач створив

конкретний плейлист. Також `tutor_id` використовується в таблицях `content`, `comments` та `likes`, щоб ідентифікувати викладача, до якого належить відповідний контент або активність.

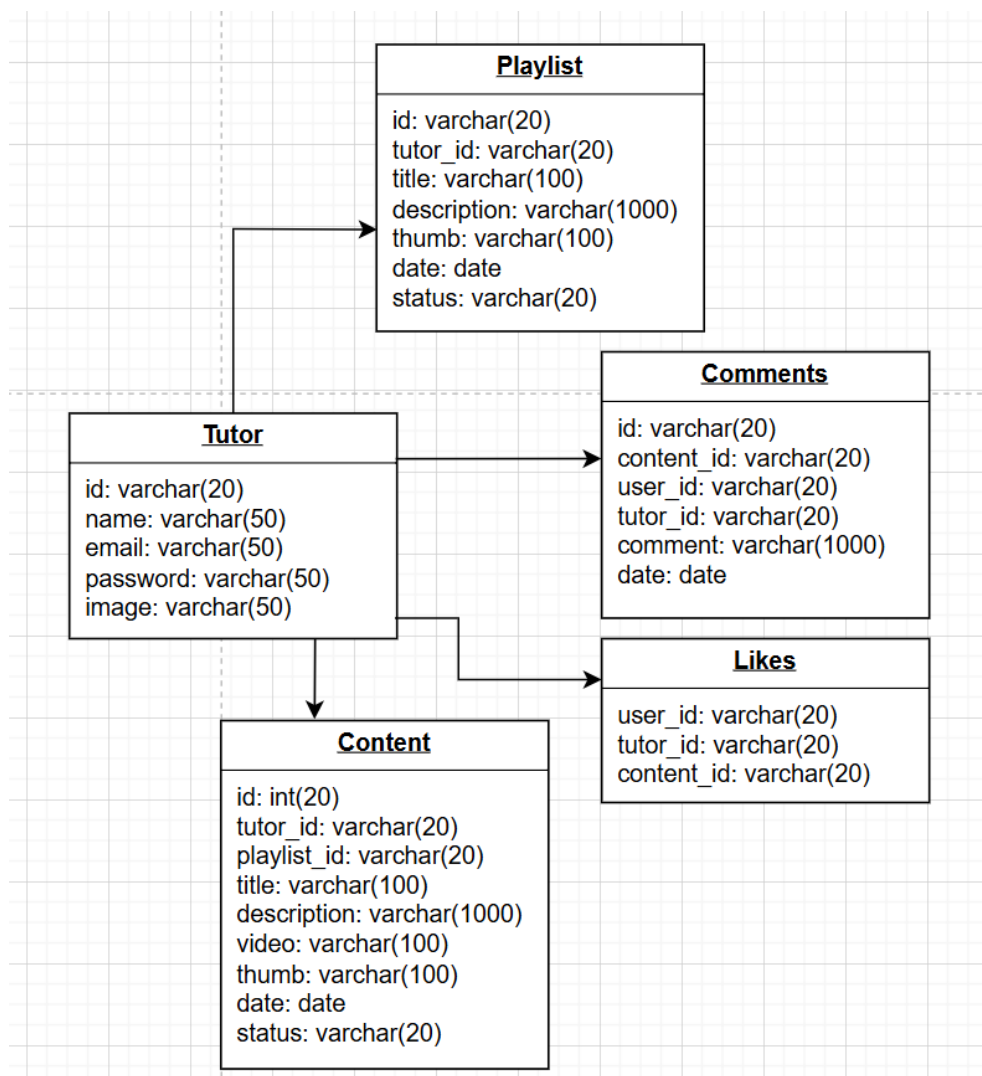


Рисунок 3.21 – Таблиця зв'язків Tutor

Таблиця `playlist` зберігає списки навчального відеоконтенту (рис. 3.22). Вона має зв'язок із таблицею `content` за допомогою поля `playlist_id`, що дозволяє групувати декілька відео в межах одного плейлиста. Також `playlist_id` використовується в таблиці `bookmark` – цей зв'язок дає змогу зберігати конкретні списки відтворення, які обрав користувач.

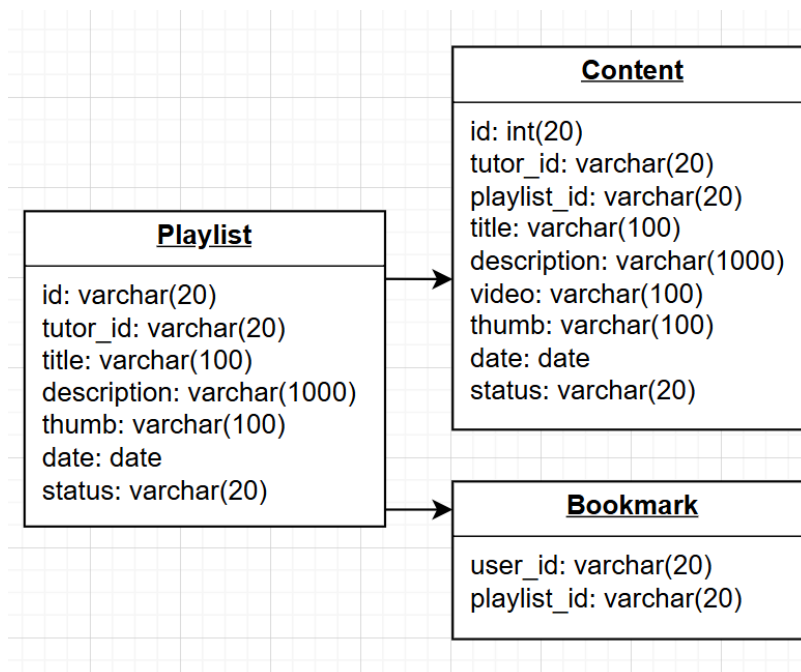


Рисунок 3.22 – Таблиця зв'язків Playlist

Таблиця content відповідає за збереження навчальних відео та пов'язаної з ними інформації. Вона має зв'язки з таблицями playlist (через playlist_id), tutor (через tutor_id), comments (через content_id) та likes (через content_id), що дозволяє встановити, до якого плейлиста, викладача, коментаря чи лайка належить той чи інший відеозапис (рис. 3.23).

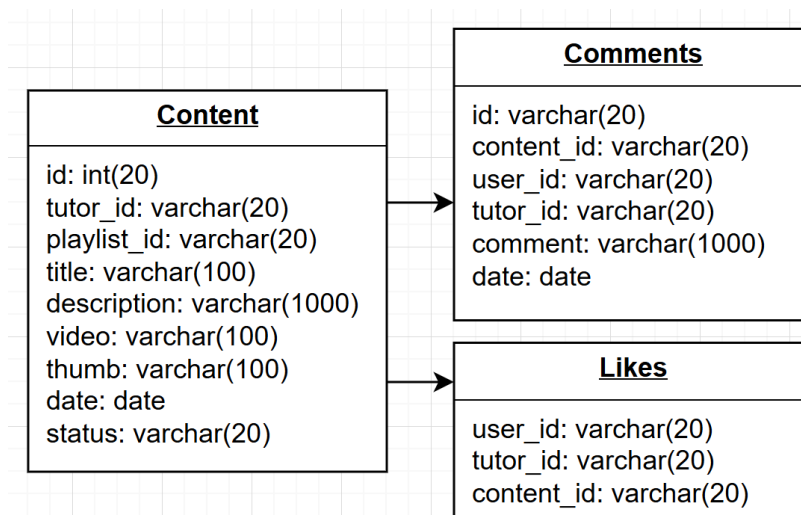


Рисунок 3.23 – Таблиця зв'язків Playlist

Таблиця comments зберігає коментарі, залишені користувачами під відео. Вона має зв'язок із таблицею content за допомогою поля content_id, що дозволяє визначити, під яким саме відео залишено коментар (рис. 3.24). Крім того, коментар пов'язується з конкретним користувачем (user_id) та викладачем (tutor_id), що дає змогу відслідковувати контекст обговорення.

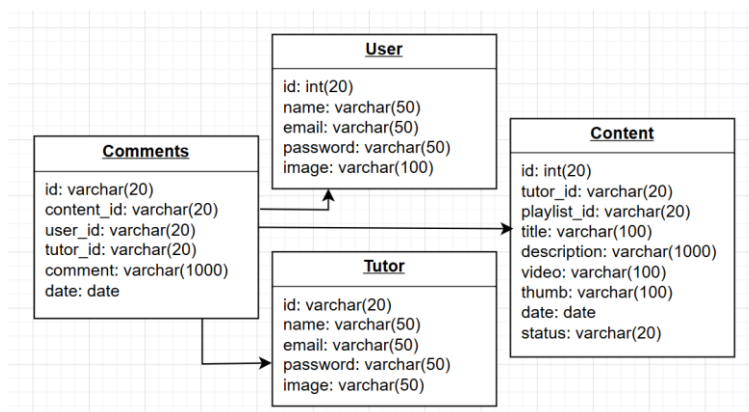


Рисунок 3.24 – Таблиця зв'язків Comments

Таблиця likes фіксує взаємодію користувачів із контентом у вигляді лайків. Вона містить поля user_id, tutor_id та content_id, які формують зв'язки з відповідними таблицями users, tutor та content, дозволяючи точно визначити, хто лайкнув яке відео та якому викладачу воно належить (рис. 3.25).

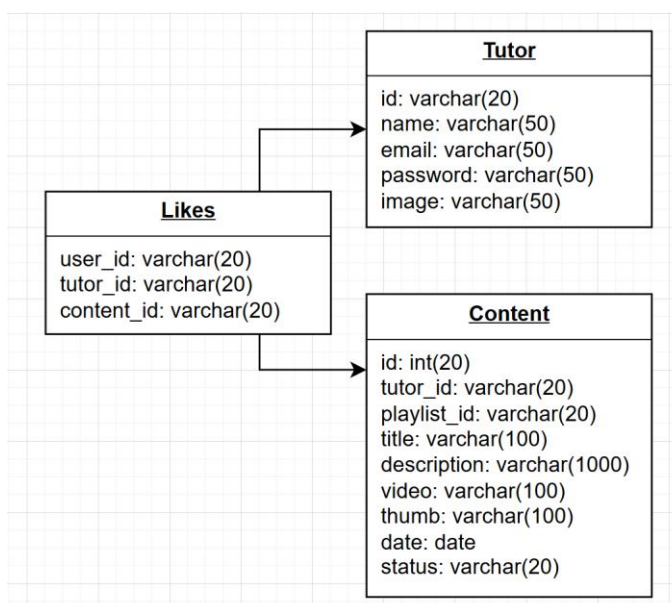


Рисунок 3.25 – Таблиця зв'язків Likes

Таблиця bookmark дозволяє користувачам зберігати улюблені плейлисти. Вона містить поля user_id та playlist_id, що формують зв'язки з таблицями users та playlist, відповідно, забезпечуючи індивідуальне збереження контенту для кожного користувача (рис. 3.26).

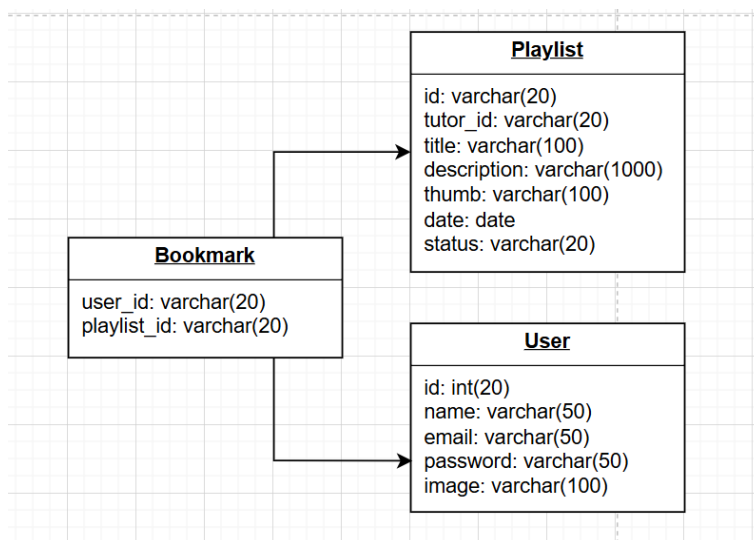


Рисунок 3.26 – Таблиця зв'язків Bookmark

Таблиця contact зберігає повідомлення, надіслані через контактну форму, включаючи ім'я, email, номер телефону та текст повідомлення. Вона є незалежною таблицею і не має зв'язків з іншими сутностями бази даних.

На основі створених ER-діаграми було сформовано SQL-запити, що описують створення таблиць бази даних із зазначенням полів, типів даних, первинних та зовнішніх ключів, а також додаткових параметрів.

Для забезпечення взаємодії між серверною частиною вебплатформи та базою даних використовується об'єктно-орієнтований підхід із застосуванням PHP і розширення PDO. Усі запити до бази даних реалізовано через централізоване з'єднання, яке ініціалізується у файлі connect.php (ліст. 3.8).

Лістинг 3.8 – Код файлу «connect.php»

```

if(isset($_POST['edit_comment'])){

    $edit_id = $_POST['edit_id'];
    $edit_id = filter_var($edit_id, FILTER_SANITIZE_STRING);
  
```

```

$comment_box = $_POST['comment_box'];
$comment_box = filter_var($comment_box, FILTER_SANITIZE_STRING);

$verify_edit_comment = $conn->prepare("SELECT * FROM `comments` WHERE
id = ? AND comment = ? ");
$verify_edit_comment->execute([$edit_id, $comment_box]);

if($verify_edit_comment->rowCount() > 0){
    $message[] = 'comment already added!';
}else{
    $update_comment = $conn->prepare("UPDATE `comments` SET comment =
? WHERE id = ? LIMIT 1");
    $update_comment->execute([$comment_box, $edit_id]);
    $message[] = 'comment updated successfully!';
}
}

```

кінець лістингу 3.8

Цей файл підключається до кожного PHP-скрипта, що працює з даними, що дозволяє уникнути дублювання коду та забезпечити зручність в обслуговуванні.

У файлі створюється новий екземпляр об'єкта PDO, який встановлює з'єднання з локальною базою даних MySQL, розміщеною на сервері localhost. Назва бази даних, яка використовується у проєкті – `course_db`, а для підключення використовується стандартний обліковий запис адміністратора з іменем користувача `root` та порожнім паролем. Завдяки використанню PDO забезпечується не лише стабільне підключення, а й підвищений рівень безпеки, оскільки всі SQL-запити виконуються з використанням підготовлених виразів, що запобігає ін'єкціям SQL.

Також у цьому ж файлі реалізовано допоміжну функцію `create_unique_id()`, яка генерує унікальні ідентифікатори для користувачів, курсів, коментарів та інших об'єктів системи. Це дозволяє уникнути дублювання даних та забезпечує цілісність записів у базі даних.

Після налаштування та успішного встановлення з'єднання з базою даних логічним наступним кроком є безпосереднє формування SQL-запитів, за допомогою яких здійснюється читання, вставка, оновлення чи видалення

інформації. Найбільш базовим і поширеним запитом, який використовується у багатьох сценаріях, є SELECT, що дозволяє отримувати дані з таблиць бази.

У межах проєкту цей запит застосовується, зокрема, для виведення списків курсів, коментарів, збережених плейлистів, інформації про користувачів та інше. Наприклад, при реалізації функції пошуку курсів запит `SELECT * FROM playlist WHERE title LIKE ...` дозволяє отримати всі записи з таблиці `playlist`, у яких заголовок містить вказане користувачем ключове слово. Такі запити завжди формуються за допомогою підготовлених виразів, що не лише захищає від SQL-ін'єкцій, а й підвищує стабільність взаємодії з базою.

Формування запитів здійснюється через об'єкт `$conn`, створений у файлі `connect.php`, цей об'єкт надає метод `prepare()`, який приймає SQL-запит у вигляді шаблону з параметрами-заповнювачами. Після підготовки запит виконується через метод `execute()`, до якого передається масив значень параметрів, у випадку читання даних результат обробляється методами `fetch`. Такий підхід використовується в усьому проєкті: від авторизації користувача до виведення контенту курсу.

3.4 Захист інформаційно-комп'ютерної системи

Захист інформації є невід'ємною складовою будь-якої сучасної онлайн-системи, особливо якщо йдеться про освітню платформу, яка оперує персональними даними користувачів. У процесі розробки було враховано ключові аспекти інформаційної безпеки як на рівні зберігання даних, так і під час їх передачі. Основні елементи інформаційної безпеки були враховані протягом усього процесу розробки, як при зберіганні даних, так і при їх передачі. Зокрема, платформа пропонує надійну систему автентифікації та авторизації, яка дозволяє розмежовувати права доступу звичайних користувачів та адміністраторів. Всі паролі, хешуються за допомогою безпечних алгоритмів, а не зберігаються у відкритому вигляді. У разі несанкціонованого доступу до бази даних цей метод усуває ймовірність компрометації справжніх даних.

Передача даних між клієнтом і сервером здійснюється із використанням захищеного з'єднання HTTPS, що унеможливорює їхнє перехоплення або модифікацію під час транспортування мережею. Особливу увагу було приділено захисту від поширених типів атак, зокрема SQL-ін'єкцій, міжсайтового скриптингу (XSS) та міжсайтових запитів (CSRF). Усі вхідні дані проходять перевірку, фільтрацію та відповідну обробку перед тим, як потрапити у базу, що істотно знижує ризики зовнішнього впливу на систему. Окрім цього, реалізовано механізми обмеження кількості невдалих спроб входу, що дозволяє захиститися від спроб підбору пароля.

Функціональність системи доповнено фіксацією дій користувачів, що дозволяє відстежувати ключові події в системі, зокрема зміну даних, додавання контенту, авторизацію та інші операції, пов'язані із безпекою. У разі інциденту цей метод полегшує аудит подій і сприяє створенню прозорого середовища. Особлива увага приділяється резервному копіюванню даних, що гарантує можливість їх відновлення в разі атаки зловмисників або технічних збоїв. Таким чином, реалізована платформа передбачає багаторівневий захист: від безпечного зберігання та передачі даних до моніторингу активності користувачів і запобігання потенційним загрозам. Комплексний підхід до безпеки дозволяє не лише захистити персональні дані, а й забезпечити стабільну роботу освітньої платформи в умовах зростаючих кіберзагроз.

3.5 SEO інформаційно-комп'ютерної системи

У сучасному цифровому світі неможливо розробити вебресурс без урахування потреб пошукової оптимізації. Ключовим елементом розробки є SEO, іншими словами пошукова система, яка визначає наскільки легко користувачам знайти вебсайт у пошукових системах, таких як Google [16]. З огляду на поширення онлайн-платформ, особливо в освітньому секторі, стає дедалі важливішим не лише створювати якісні ресурси, а й забезпечувати їх ефективне просування та доступність в Інтернеті [17].

SEO-оптимізація охоплює як технічні, так і змістовні елементи. Вона передбачає розробку вебсайту з чіткою структурою, правильне використання заголовків і метаданих, адаптацію інтерфейсу до різних типів пристроїв, скорочення часу завантаження сторінок і надання користувачам релевантного, корисного контенту [18]. Усе це допомагає сайту займати вищі позиції в результатах пошуку за ключовими словами, які потенційні користувачі можуть вводити в пошуковий рядок, і покращує його індексацію пошуковими системами.

У контексті платформ онлайн-навчання, де критично важливо налагодити зв'язок з якомога більшою кількістю студентів, викладачів та інших зацікавлених сторін, SEO має особливе значення. Ефективна SEO-стратегія гарантує більшу видимість, зростання трафіку і, як наслідок, активніше використання платформи за призначенням для таких ресурсів. Ще до того, як користувач відвідає вебсайт, коли він ще переглядає результати пошуку, завдяки правильній оптимізації можна встановити з ним сприятливий перший візуальний і функціональний зв'язок.

У рамках проєкту вебплатформи «Educa.» було впроваджено основні принципи SEO-оптимізації, що включають налаштування ключових тегів, формування логічних URL-адрес, покращення швидкості завантаження, а також адаптацію контенту до потреб пошукових алгоритмів. Застосування таких підходів сприяє підвищенню видимості ресурсу у пошукових системах, що, в свою чергу, позитивно впливає на його популярність, охоплення цільової аудиторії та загальну ефективність використання у навчальному процесі.

Висновки по розділу 3

У процесі реалізації освітньої платформи було здійснено повний цикл її розробки та впровадження, що охоплює проєктування інтерфейсів, реалізацію функціоналу, тестування, розробку бази даних, заходи із захисту інформації та оптимізацію пошукових систем.

Проведене тестування дозволило виявити помилки, перевірити стабільність роботи та забезпечити відповідність системи вимогам до якості та зручності.

Особливу увагу приділено структурі бази даних, яка оптимізована для зберігання освітнього контенту, обліку користувачів та взаємодії між ними. Було здійснено реалізацію фізичної моделі бази даних, яка охоплює основні сутності системи, усі таблиці пов'язані між собою відповідними ключами, що забезпечує логічну цілісність даних і стабільну роботу всієї системи.

На рівні безпеки впроваджено низку заходів, серед яких – підготовлені SQL-запити, хешування паролів, перевірка сесій, фільтрація вхідних даних і контроль завантажуваних файлів. Усе це забезпечує базовий рівень захисту даних та доступу до системи.

Також багато уваги було приділено SEO-оптимізації: було впорядковано структуру URL-адрес, змінено метадані, забезпечено гнучкість дизайну та зменшено швидкість завантаження сторінок. Це збільшує відвідуваність сайту, робить його більш помітним у пошуковій видачі та гарантує ефективне залучення цільової аудиторії.

Таким чином, створена освітня платформа відповідає основним вимогам до сучасного вебпродукту, є придатною для практичного використання й може бути розширена та вдосконалена в майбутньому.

ВИСНОВКИ

У ході реалізації проєкту було успішно створено онлайн-платформу для освітнього процесу з використанням JavaScript та PHP, що підтверджує виконання всіх технічних і функціональних вимог.

Було здійснено аналіз сучасного стану розвитку онлайн-освіти та проведено аналіз відомих онлайн-платформ, а саме Prometheus, EdEra, Дія.Освіта, Kwigа та Coursera. Це стало основою для визначення ключових вимог до розроблюваної онлайн-платформи.

Після аналізу наявних платформ, було визначено функціональні вимоги, які мають бути реалізовані в системі для двох основних типів користувачів, а саме взаємодія з контентом та плейлистами з сторони студента та адміністратора, перегляд власної аналітики, редагування та створення курсу та його наповнення викладачем, та взаємодія студента з цим плейлистом.

На основі зібраних вимог було сформоване уявлення про загальну архітектуру майбутнього програмного продукту. Для наочного представлення структури та логіки взаємодії компонентів системи було застосовано структурно-об'єктну модель із використанням нотації UML.

Для розробки було обрано набір технологій, що дозволило реалізацію проєкту серед яких є мова розмітки HTML, мова стилів CSS, мова програмування JavaScript, серверна мова програмування PHP, система керування базами даних MySQL, а також середовище локального розгортання XAMPP. Саме поєднання цих технологій дозволило створити повноцінну багатофункціональну систему з чітким поділом на візуальну частину й серверну логіку.

На етапі розробки було безпосередньо реалізовано функціональну частину платформи згідно з вимогами та логікою, визначеними під час моделювання. Роботу поділено на дві основні частини: серверну (бекенд), реалізовану мовою програмування PHP, та клієнтську (фронтенд), реалізовану засобами HTML, CSS і JavaScript.

Після реалізації основної функціональності було проведено всебічне тестування платформи. Основним підходом стало ручне тестування, у межах якого перевірялися всі ключові сценарії взаємодії, також для кращої ефективності перевірки використовувалися засоби штучного інтелекту (ШІ). За допомогою LLM-моделей проводився аналіз коду на предмет потенційних помилок, неефективностей, вразливостей у логіці перевірки та обробки введення.

Для забезпечення взаємодії між серверною частиною вебплатформи та базою даних використовується об'єктно-орієнтований підхід із застосуванням PHP і розширення PDO. Було здійснено реалізацію фізичної моделі бази даних, яка охоплює основні сутності системи, усі таблиці пов'язані між собою відповідними ключами, що забезпечує логічну цілісність даних і стабільну роботу всієї системи.

На рівні безпеки впроваджено низку заходів, серед яких – підготовлені SQL-запити, хешування паролів, перевірка сесій, фільтрація вхідних даних і контроль завантажуваних файлів. Усе це забезпечує базовий рівень захисту даних та доступу до системи.

Значну увагу приділено SEO-оптимізації, було реалізовано структуру URL-адрес, налаштовано метадані, забезпечено адаптивність дизайну, зменшено час завантаження сторінок. Це дозволяє покращити видимість ресурсу у пошукових системах, сприяє зростанню відвідуваності та забезпечує ефективне залучення цільової аудиторії.

Створена онлайн-платформа для освітнього процесу повністю відповідає поставленим вимогам та забезпечує зручну взаємодію користувачів із освітнім ресурсом, надійність та безпеку даних. Платформа готова до подальшого розвитку та масштабування, гарантуючи високу якість розробленого продукту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Qinyi Tan Q. T. Increasing Public Interest in Online Education during the COVID-19 Pandemic in the United States: An Analysis of Google Trends Data. URL: https://www.researchgate.net/publication/387507139_Increasing_Public_Interest_in_Online_Education_during_the_COVID-19_Pandemic_in_the_United_States_An_Analysis_of_Google_Trends_Data (date of application: 19.02.2025).
2. The COVID-19 pandemic has changed education forever. This is how. World Economic Forum. URL: <https://www.weforum.org/stories/2020/04/coronavirus-education-global-covid19-online-digital-learning/> (date of application: 19.02.2025).
3. Офіційний сайт Міністерства освіти і науки України. URL: <https://mon.gov.ua/> (дата звернення: 20.02.2025).
4. Державна служба якості освіти України. URL: <https://sqe.gov.ua/> (дата звернення: 20.02.2025).
5. Prometheus. URL: <http://prometheus.org.ua> (дата звернення: 20.02.2025).
6. EdEra. URL: <https://ed-era.com/> (дата звернення: 20.02.2025).
7. Дія.Освіта. URL: <https://osvita.diia.gov.ua/> (дата звернення: 21.02.2025).
8. KWIGA. URL: <https://kwiga.com/ua> (дата звернення: 21.02.2025).
9. Coursera. URL: www.coursera.org/ (дата звернення: 21.02.2025).
10. HTML: HyperText Markup Language. MDN Web Docs. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML> (date of application: 03.03.2025).
11. CSS: Cascading Style Sheets. MDN Web Docs. URL: <https://developer.mozilla.org/en-US/docs/Web/CSS> (date of application: 04.03.2025).
12. JavaScript. MDN Web Docs. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (date of application: 04.03.2025).
13. W3Schools.com. Online Web Tutorials. URL: https://www.w3schools.com/php/php_intro.asp (date of application: 04.03.2025).
14. MySQL. URL: <https://dev.mysql.com/doc/> (date of application: 05.03.2025).

15. XAMPP Installers and Downloads for Apache Friends. URL: <https://www.apachefriends.org/index.html> (date of application:05.03.2025).

16. Beginner's Guide to SEO. URL: <https://moz.com/beginners-guide-to-seo> (date of application:01.05.2025).

17. SEO Starter Guide: The Basics. Google Search Central. Documentatio. Google for Developers. URL: <https://developers.google.com/search/docs/fundamentals/seo-starter-guide> (date of application: 23.04.2025).

18. What Is SEO. URL: <https://neilpatel.com/what-is-seo/> (date of application: 23.04.2025).

ДОДАТКИ

Додаток А

Підтвердження апробації результатів кваліфікаційної роботи

Міністерство освіти і науки України
 Волинська обласна рада
 Луцька міська рада
 Луцький національний технічний університет
 Національний технічний університет України «Київський політехнічний
 інститут імені Ігоря Сікорського»
 Національний університет «Львівська політехніка»
 Військовий інститут телекомунікацій та інформатизації
 імені Героїв Крут (м. Київ)
 Тернопільський національний педагогічний університет імені В. Гнатюка
 Рівненський державний гуманітарний університет
 Навчально-науковий інститут «Українська інженерно-педагогічна академія»
 Харківського національного університету ім. В.Н. Каразіна
 Люблінська політехніка (Польща)
 Фрайберзька гірнича академія (Німеччина)
 Гронінгенський університет (Нідерланди)
 Кавказький університет (Грузія)
 Політехнічний університет Браганси (Португалія)



ТЕЗИ ДОПОВІДЕЙ

**X Міжнародної науково-практичної конференції з
 проблем вищої освіти і науки «Інформаційні технології
 в освіті, науці і виробництві (ІТОНВ-2025)»**

23-24 травня 2025 р.

Луцьк 2025

Shumyliak Liliia	Implementation of web technologies in	271
Cibák Luboš	the management of medical department	
Tarnovetska Olha	workflows	

СЕКЦІЯ 5. ІНФОРМАЦІЙНІ ТА ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ І ТЕХНОЛОГІЇ

Антонюк А.О.	Голосовий помічник на основі	275
Повстяна Ю.С.	штучного інтелекту	
Вовк П.Б.	Підготовка середовища та налаштування локального кластера Kubernetes: від теорії до практики	282
Вознюк А.В.	Розробка 2D ігор на unity з генерацією	286
Данилко К.С.	світу	
Вознюк М.В.	Аналіз впливу темної теми на зручність	289
Неділько О.В.	користування веб-додатками	
Вознюк А.В.	Автоматизація сервісів доставки їжі за	293
Хітько Ю.В.	допомогою Telegram-бота з AI та платіжним модулем	
Вознюк А.В.	Технічні рішення та освітні переваги	298
Чоботар Ю.С.	при створенні і впровадженні	
Примич М.О.	вебплатформи для онлайн-навчання	
Газдюк К.П.	Адаптивне планування логістичних	304
Дмитрашук К.М.	маршрутів в умовах воєнного стану	
Кривинчук В.О.		
Головня С.А.	Застосування сучасних бібліотек Python	308
	для задач з аналізу та прогнозування	
Демідов П.Г.	Порівняльна характеристика нейро-	312
Андрієнко В.А.	нечітких та нейронних технологій на	
Маргаза Д.Ю.	прикладі розв'язання задачі	
Муляр В.П.	прогнозування вартості валюти на фінансовому ринку	

УДК 004.85

ТЕХНІЧНІ РІШЕННЯ ТА ОСВІТНІ ПЕРЕВАГИ ПРИ СТВОРЕННІ І ВПРОВАДЖЕННІ ВЕБПЛАТФОРМИ ДЛЯ ОНЛАЙН-НАВЧАННЯ

Вознюк Анастасія Вадимівна

Луцький національний технічний університет, асистент кафедри інженерія програмного забезпечення, a.vozniuk@lutsk-ntu.com.ua

Чоботар Юлія Сергіївна

Луцький національний технічний університет, студентка 4 курсу кафедри інженерія програмного забезпечення, chobotar.ly@gmail.com

Примич Марія Олександрівна

Луцький національний технічний університет, студентка 4 курсу кафедри інженерія програмного забезпечення, mashameky@gmail.com

TECHNICAL SOLUTIONS AND EDUCATIONAL BENEFITS IN THE DEVELOPMENT AND IMPLEMENTATION OF AN ONLINE LEARNING PLATFORM

Vozniuk Anastasiia Vadymivna

Lutsk National Technical University, Assistant at the Department of Software Engineering, a.vozniuk@lutsk-ntu.com.ua

Chobotar Yulia Serhiivna

Lutsk National Technical University, Student at the Department of Software Engineering, chobotar.ly@gmail.com

Prymych Maria Alexandrovna

Lutsk National Technical University, Student at the Department of Software Engineering, mashameky@gmail.com

The development of an online educational platform is presented, focusing on modern trends in distance learning accelerated by the COVID-19 pandemic. It has been established that such platforms provide flexible, accessible, and cost-effective learning opportunities. The integration of video courses and gamification elements enhances user engagement and supports self-paced learning. An experiment by the Ukrainian company CleverStaff demonstrates the effectiveness of gamified approaches in improving motivation. The study also outlines the main challenges of online education and emphasizes the growing relevance of hybrid models, microlearning, and AI-powered personalization.

Keywords: online education, distance learning, gamification, video courses, COVID-19 pandemic, digital transformation, web platforms.

Пандемія COVID-19 суттєво вплинула на всі сфери життя, зокрема й на освіту. Із запровадженням карантинних обмежень

мільйони учнів, студентів і викладачів по всьому світу були змушені у стислі терміни перейти на дистанційне навчання. Це стало поштовхом до стрімкого розвитку онлайн-освіти, яка до того моменту залишалася переважно альтернативною або додатковою формою навчання. Онлайн-платформи почали масово впроваджуватися в шкільній, університетській та післядипломній освіті, а технології відеозв'язку, інтерактивного контенту, хмарних сервісів і мобільних застосунків стали звичним інструментом освітнього процесу.

Зростання популярності онлайн-навчання добре ілюструє графік [1] частотності пошукових запитів за ключовими словами “online school”, “online education”, “online teaching” і “online learning” у період із 2019 по 2023 рік.

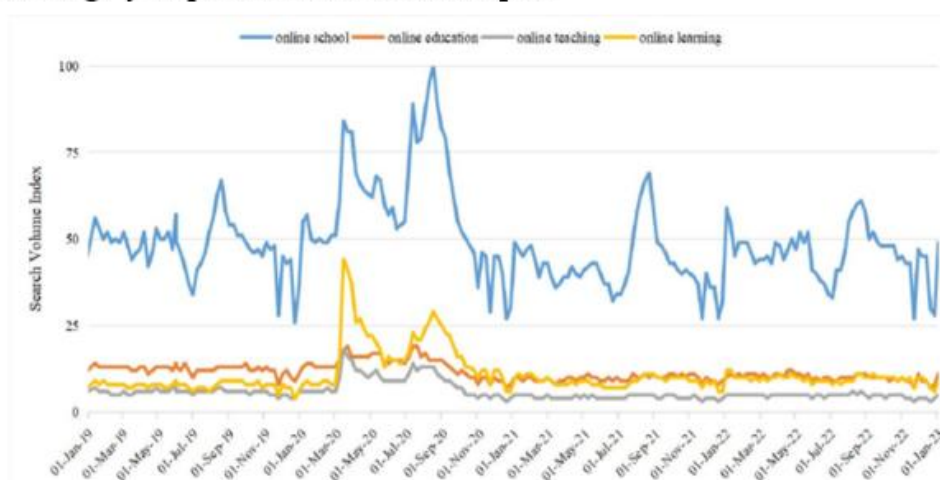


Рисунок 1 – Графік популярності пошукових запитів 2019-2023 років

На ньому чітко видно, що найвищий інтерес спостерігався на початку 2020 року, що прямо пов'язано з початком пандемії. Найпомітніше зростання демонструє запит “online school”, що цілком логічно, адже саме школи одними з перших були переведені в онлайн-формат. Інші запити також демонструють зростання, але менш виражене. Після піку у 2020 році рівень зацікавленості дещо знизився, однак залишився на стабільно

вищому рівні, ніж до пандемії, що свідчить про те, що онлайн-освіта закріпилася як невіддільна частина освітньої екосистеми.

Однією з головних причин популярності онлайн-формату стало те, що він забезпечує гнучкість і доступність: користувачі можуть навчатись у зручний для себе час і місці, маючи доступ до освітніх матеріалів з будь-якої точки світу. Такий формат дозволяє поєднувати навчання з іншими аспектами життя — роботою, хобі, сім'єю.

Також доступність онлайн освіти дає змогу людям з обмеженнями розвиватись прямо з дому і отримувати бажані знання уникаючи труднощів традиційного офлайн навчання.

Кожен студент чи учень має змогу підібрати для себе застосунки з різними методами освоєння інформації та можливістю обрати зручний час для навчання, повторення матеріалу. Крім того, цифрове навчання дозволяє значно зменшити витрати, пов'язані з поїздками, проживанням і друком матеріалів.

Під час пандемії активно почали розвиватись вебзастосунки та мобільні застосунки, які зосереджені на залученості та мотивації користувача.

Одним із ключових елементів сучасної онлайн-освіти стали відеокурси, які суттєво змінили підхід до подачі навчального матеріалу. Завдяки можливості запису лекцій у зручному форматі, викладачі можуть охопити значну аудиторію без необхідності постійної присутності. Відеоуроки дозволяють студентам переглядати матеріал у будь-який зручний для них час, ставити відео на паузу, повертатися до складних моментів і засвоювати інформацію у власному темпі.

Однак для ще більшого залучення та утримання уваги користувачів, освітні платформи дедалі частіше впроваджують додаткові механізми, зокрема гейміфікацію, яка робить навчання більш захопливим та цікавим, перетворюючи його на своєрідну гру, що сприяє активній участі та позитивним емоціям.

За рахунок аспектів внутрішньої, зовнішньої мотивації, автономії, відчуття компетентності та соціальних взаємодій у 2022 році українська компанія CleverStaff провели невеликий експеримент розробивши ексклюзивне оновлення – систему гейміфікації [5]. Її ціль — поліпшити взаємодію користувачів та рекрутерів через зацікавлення у тій же роботі. За кілька місяців після релізу команда розробників проаналізувала, що змінилось, коли рекрутери почали застосовувати гейміфікований софт. Результати опитування зображені на діаграмі [2].

Що вам більше за все сподобалося?



Рисунок 2 – Діаграма опитування серед працівників CleverStaff

Результати підтвердили, мотивація зростає через нагородження чи отримання призового місця серед інших користувачів що забезпечує ефективність у справах.

Водночас зросла можливість масштабування навчання, адже один онлайн-курс можуть проходити тисячі людей одночасно. А також онлайн-освіта здатна допомогти вирішити питання нестачі викладачів та навчальних приміщень.

Разом з тим, онлайн-освіта має і свої недоліки. Відсутність живої комунікації може впливати на якість взаємодії між студентами та викладачами, а також ускладнювати соціалізацію. Освіта у Інтернеті може призводити до відчуття самотності та зниження зацікавленості до навчання.

Для багатьох користувачів проблемою є низька самодисципліна або брак мотивації. Не всі мають рівний доступ до якісного інтернету чи необхідної техніки, крім того, не всі педагоги достатньо підготовлені до ефективного використання цифрових технологій, що може впливати на якість подачі матеріалу.

Сьогодні онлайн-освіта продовжує розвиватися, трансформуючись у гібридні моделі, які поєднують переваги очного та дистанційного навчання. Зростає популярність короткострокових сертифікаційних курсів, мікронавчання, адаптивних платформ з елементами штучного інтелекту та автоматизованими системами підтримки, що надає змогу аналізувати дані про прогрес студента, його сильні та слабкі моменти. Усе це свідчить про те, що онлайн-освіта перестала бути тимчасовим рішенням і набула стратегічного значення в контексті модернізації всієї освітньої системи.

Онлайн-освіта з тимчасового рішення у період пандемії перетворилася на повноцінний і перспективний формат навчання, що продовжує активно розвиватися. Її ключові переваги — гнучкість, доступність і масштабованість — роблять цифрове навчання актуальним інструментом для забезпечення якісної освіти в умовах діджиталізації. Проте для підвищення ефективності та мотивації користувачів важливо не просто переносити контент у цифрове середовище, а адаптувати його до нових форматів взаємодії.

Саме тому особливо доцільною є розробка онлайн-курсів, що поєднують відеоуроки та елементи гейміфікації. Відео дає змогу навчатись у власному темпі, повертатися до складного матеріалу та краще сприймати інформацію завдяки візуалізації. Гейміфікація ж стимулює інтерес і залученість, сприяє формуванню внутрішньої мотивації через систему досягнень, рейтинги, нагороди чи інші інтерактивні механізми. Такий підхід дозволяє зробити освітній процес не лише доступним, а й дійсно ефективним — особливо в умовах, коли учням і студентам часто бракує самодисципліни чи зовнішніх стимулів.

Інвестування в розробку подібних платформ є виправданим і стратегічно важливим кроком, адже воно сприяє як підвищенню якості освіти, так і розширенню можливостей для здобуття знань незалежно від місця проживання, соціального статусу чи фізичних можливостей. Онлайн-навчання з відеоуроками та гейміфікацією — це відповідь на сучасні освітні виклики та реальна перспектива для побудови ефективної, привабливої й доступної освіти майбутнього.

Список використаних джерел

1. Increasing Public Interest in Online Education during the COVID-19 Pandemic in the United States: An Analysis of Google Trends Data. URL: https://www.researchgate.net/publication/387507139_Increasing_Public_Interest_in_Online_Education_during_the_COVID-19_Pandemic_in_the_United_States_An_Analysis_of_Google_Trends_Data (дата звернення: 29.04.2025).
2. The COVID-19 pandemic has changed education forever. URL: https://www.weforum.org/stories/2020/04/coronavirus-education-global-covid19-online-digital-learning/?utm_source=chatgpt.com (дата звернення: 13.04.2025).
3. Офіційний сайт Міністерства освіти і науки України. URL: <https://mon.gov.ua/> (дата звернення: 13.04.2025)
4. Державна служба якості освіти України. URL: <https://sqe.gov.ua/> (дата звернення: 13.04.2025)
5. Як гейміфікація впливає на ефективність роботи. URL: <https://happymonday.ua/gejmifikatsiya-vplyvaye-na-robotu-rekrutera> (дата звернення: 27.04.2025)