

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ ДЛЯ
ПРОХОДЖЕННЯ ОНЛАЙН КУРСІВ
SOFTWARE IMPLEMENTATION OF A WEB
APPLICATION FOR ONLINE COURSES

Виконав(ла): студент(ка) 4 курсу, групи КНТ-151

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

МОРОЗОВ К.Д.

(ПРИЗВИЩЕ та ініціали)

Керівник ОЛІЙНИК А.О.

(ПРИЗВИЩЕ та ініціали)

Рецензент СКРУПСЬКИЙ С.Ю.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Факультет КНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 121 Інженерія програмного забезпечення
 (код і найменування)
 Освітня програма (спеціалізація) Інженерія програмного забезпечення
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
 “ ” 2025 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

МОРОЗОВ Кирило Дмитрович
 (ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмна реалізація вебзастосунку для проходження онлайн курс Software implementation of a web application for online courses

керівник проєкту (роботи) к.т.н., професор, Олійник Андрій Олександрович,
 (науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 28 ” квітня 2025 року № 209

2. Строк подання студентом проєкту (роботи) 02 червня 2025 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ОЛІЙНИК А.О., професор		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання « 14 » квітня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій	2 тиждень	Розділ 2
	розробки.		
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділ 3, 4
6	Тестування та експериментальне дослідження	5 тиждень	Розділ 4
	програмного забезпечення.		
7	Оформлення пояснювальної записки та документів	6 тиждень	Додатки
	до неї.		
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис) Кирило МОРОЗОВ
(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Андрій ОЛІЙНИК
(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 86 с., 15 табл., 52 рис., 3 дод., 34 джерел.

ПЛАТФОРМА ОНЛАЙН НАВЧАННЯ, ОНЛАЙН КУРСИ, NODEJS, REACTJS, NEXTJS, NESTJS, TYPESCRIPT, JAVASCRIPT.

Об'єкт дослідження – процес розробки програмного забезпечення для онлайн навчання.

Предмет дослідження – вебплатформи для дистанційного навчання.

Мета роботи – створення онлайн платформи, що забезпечує ефективне навчання, взаємодію викладачів і студентів, управління навчальним процесом та автоматизацію освітніх процесів.

Матеріали, методи та технічні засоби: структурне та об'єктно-орієнтоване програмування, вебтехнології (HTML, CSS, JavaScript), серверні технології (Node.js, NestJS), бази даних (PostgreSQL), фреймворки React та Next.js.

Результати. Розроблено вебплатформу для онлайн навчання, що включає систему управління курсами, тестування, аналітику успішності та інтеграцію з відеоконференціями.

Висновки. Створена система дозволяє автоматизувати навчальний процес, покращити доступність та ефективність онлайн освіти, забезпечуючи зручний інтерфейс та широкий набір функцій для користувачів.

Галузь використання – заклади освіти, корпоративне навчання, онлайн курси та самостійне навчання.

ABSTRACT

The bachelor's qualification thesis explanatory note: 86 pages, 15 tables, 52 figures, 3 appendices, 34 sources.

ONLINE LEARNING PLATFORM, ONLINE COURSES, NODE.JS, REACT.JS, NEXT.JS, NESTJS, TYPESCRIPT, JAVASCRIPT.

Object of research – process of developing software for online learning.

Subject of research – web platforms for distance learning.

Purpose of the study – to develop an online platform that ensures effective learning, interaction between teachers and students, management of the educational process, and automation of learning activities.

Materials, methods, and technical tools: structured and object-oriented programming, web technologies (HTML, CSS, JavaScript), server-side technologies (Node.js, NestJS), databases (PostgreSQL), frameworks (React, Next.js), and cloud deployment services (AWS, Vercel).

Results. A web-based learning platform has been developed, featuring course management, testing, performance analytics, and integration with video conferencing tools.

Conclusions. The developed system enables automation of the learning process, enhances accessibility and effectiveness of online education, and provides a user-friendly interface with a comprehensive set of features.

Field of application – educational institutions, corporate training, online courses, and self-paced learning.

ЗМІСТ

	С.
Перелік скорочень та умовних познач	7
Вступ	8
1 Аналіз предметної області	9
1.1 Огляд існуючих рішень	9
1.2 Постанова завдання	20
1.3 Висновки за розділом 1	26
2 Матеріали і методи	28
2.1 Вибір мови програмування	28
2.2 Вибір середовища розробки	29
2.3 Вибір фреймворку	30
2.4 Вибір бази даних та orm	32
2.5 Розробка моделі взаємодії користувача	33
2.6 Висновки за розділом 2	35
3 Опис та реалізація програми	37
3.1 Вимоги до технічних і програмних засобів	37
3.2 Архітектура програмного забезпечення	37
3.3 Розробка модулів системи	39
3.4 Розробка алгоритмів	40
3.4 Проєктування бази даних	43
3.5 Розробка користувацького інтерфейсу	46
4 Експлуатація, тестування та експериментальне дослідження програми	64
4.1 Опис роботи з програмою	64
4.2 Тестування програмного забезпечення	68
Висновки	71
Перелік джерел посилання	72
Додаток А Технічне завдання	75
Додаток Б текст програми	80
Додаток В слайди презентації	82

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

- API – Application Programming Interface;
APP – Application;
IDE – Integrated Development Environment;
ORM – Object-Relational Mapping;
ПЗ – Програмне забезпечення.

ВСТУП

У сучасному світі онлайн освіта набуває все більшої популярності завдяки своїй доступності, гнучкості та ефективності. Зростання попиту на дистанційне навчання стимулює розвиток різноманітних платформ для проведення онлайн курсів, що дозволяє студентам та професіоналам здобувати нові знання та навички без прив'язки до місця чи часу.

Актуальність розробки платформи онлайн курсів обумовлена зростаючою потребою у доступних та гнучких інструментах навчання для студентів, людей, які шукають перше місце роботи, а також тих, хто прагне підвищити свою кваліфікацію. У сучасних умовах ринку праці важливо мати можливість швидко здобувати нові знання та навички, що підвищують конкурентоспроможність фахівців. Попри наявність відомих платформ, таких як UdeMy [1] чи Coursera [2], багато користувачів потребують рішень, що краще відповідають їхнім індивідуальним цілям і дозволяють ефективно організувати навчальний процес. Тому розробка інноваційної та зручної платформи для онлайн навчання є важливою та актуальною для підтримки професійного розвитку та успішної інтеграції на ринку праці.

Об'єктом дослідження є процес створення вебзастосунку для онлайн курсів, що забезпечує ефективну взаємодію між викладачами та студентами.

Предметом дослідження є методи та технології розробки вебзастосунку для онлайн освіти, включаючи архітектуру платформи, функціональні можливості та засоби інтеграції з іншими сервісами.

Метою роботи є розробка та програмна реалізація вебзастосунку для онлайн курсів, що забезпечить зручний доступ до навчального контенту, можливість адміністрування курсів, інтерактивність та автоматизацію освітнього процесу.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Сучасні технології дозволяють покращити процеси навчання, роблячи навчання більш доступним та ефективним. Завдяки онлайн курсам освітній процес надає можливість користувачам отримувати знання у зручний для них спосіб.

У цьому розділі проведено аналіз предметної області, сформульовано основні поняття та визначено вимоги до розроблюваної платформи онлайн курсів.

1.1 Огляд існуючих рішень

Проаналізувавши наявні платформи для онлайн навчання, було визначено три найбільш популярних, які характеризуються найширшим спектром функціональних можливостей, а саме: Udemu [1], Coursera [2], Educative [3].

У подальшому розглядається детальний аналіз кожної з цих платформ. Аналіз представлений у наступному форматі:

Спершу представлено історичну довідку, де зазначається дата заснування, засновники та ключові фактори, що сприяли зростанню популярності платформи.

Далі подано характеристику функціональних можливостей, яка включає унікальні особливості, що вирізняють її серед аналогів.

На завершення здійснюється узагальнене оцінювання платформи, яке охоплює аналіз дизайну системи та зручності її використання.

1.1.1 Аналіз платформи онлайн навчання: Udemu

Онлайн платформа Udemu [1], була заснована у США в 2010 році. Платформа стала популярною завдяки своїм особливостям та інноваційній

для того часу моделі взаємодії між викладачами та студентами. Udemu [1] надавала можливість кожному охочому створювати навчальні курси без необхідності підтвердження кваліфікації сертифікатами або іншими офіційними документами. Не зважаючи на це, рівень якості матеріалів залишався високим завдяки внутрішній конкуренції між авторами та ефективній рейтинговій системі, яка оцінювала курси за різними критеріями, що підвищувало довіру користувачів до платформи. Ще одним важливим фактором стала гнучка модель доступу до курсів. Користувачі могли одноразово придбати курс, та отримати довічний доступ до нього, що дозволяло користувачам навчатися у власному темпі, без прив'язки до жорстких планів проходження курсу. Це виділяло Udemu [1] від платформ де користувачі мали оформити підписку для проходження курсів, та сприяло формуванню внутрішнього ринку освітнього контенту. Крім того, така стратегія дозволила платформі впроваджувати масштабні знижки, що відіграло важливу роль у залученні нових користувачів на початкових етапах її розвитку. Стремління охопити широку аудиторію також стало одним із головних пріоритетів платформи, що зумовило підтримку понад 75 мов та широкий вибір навчальних курсів з різних дисциплін. Сукупність цих факторів сприяла швидкому зростанню Udemu [1] та її перетворенню на одну з провідних освітніх платформ у світі.

Проаналізувавши систему було виявлено наступний функціонал який віділяє пталформу серед інших та який можна поділити на декілька груп:

Функціональна група для студентів:

Довічний доступ до курсів – навчальні матеріали купуються одноразово та залишаються у доступі назавжди. Платформа дозволяє студентам навчатися у власному ритмі без обмежень по часу, що робить Udemu [1] привабливим вибором для тих, хто хоче здобувати нові знання у зручний для себе час.

Офлайн навчання – офлайн перегляд курсів забезпечує можливість завантажувати навчальні матеріали через мобільний застосунок, що дозволяє

користувачам переглядати їх без доступу до мережі. Це особливо корисно для тих, хто подорожує або має нестабільне інтернет-з'єднання.

Розширена система пошуку та фільтрації – потужна система фільтрації та пошуку курсів дає змогу студентам ефективно знаходити необхідні навчальні матеріали серед великої кількості доступних курсів. Платформа пропонує розширені можливості фільтрації за категоріями та темами, рівнем складності, тривалістю навчання, мовою курсу, рейтингами, відгуками, а також за ціною, що дозволяє користувачам легко знаходити відповідні курси відповідно до своїх потреб та рівня підготовки. Для корпоративних клієнтів передбачена фільтрація за навичками, що допомагає бізнесам обирати курси, релевантні до професійного розвитку співробітників.

Кросплатформеність та мобільний доступ – Udemu [1] забезпечує зручність навчання на різних пристроях, включаючи настільні комп'ютери, ноутбуки, планшети та смартфони. Платформа доступна через сучасні браузері, такі як Chrome, Firefox, Edge та Safari, а також через офіційні мобільні додатки для iOS та Android. Адаптивний дизайн дозволяє комфортно використовувати платформу на будь-якому екрані, а синхронізація прогресу між пристроями гарантує, що користувачі можуть почати навчання на одному пристрої та безперешкодно продовжити на іншому, зберігаючи всі дані про переглянуті лекції та виконані завдання.

Функціональна група для викладачів:

Аналітика та метрики – система графіків, яка дозволяє відстежувати статистику стосовно продажу курсів, переглядів, відгуків і залученості студентів. Викладачі можуть аналізувати ефективність своїх курсів, оцінювати динаміку продажів та взаємодію студентів з матеріалами.

Гнучка система цін та знижок – надає можливість викладачам самостійно встановлювати ціни для власних курсів, брати участь у глобальних розпродажах Udemu [1] або пропонувати персоналізовані знижки для своєї аудиторії.

Інструменти для створення контенту – Udemу [1] надає вбудований конструктор для створення курсів, який містить можливість додавати відео, тест, завдання та додаткові матеріали. Викладачі можуть структурувати контент за модулями та темами, використовуючи різні формати навчальних матеріалів.

Система взаємодії зі студентами – викладачі можуть відповідати на запитання студентів, проводити обговорення в спеціальних форумах курсу та надсилати анонси щодо оновлень курсу або важливих новин.

Механізм сертифікації – після завершення курсу студенти отримують сертифікати про проходження, що підвищує цінність курсів та стимулює залученість студентів.

Корпоративне навчання для бізнесу:

Аналітика та метрики стосовно прогресу кожного працівника бізнесу – фільтрація по навчальних досягненнях співробітників, що дозволяє компаніям відстежувати прогрес. Компанії можуть отримувати детальні звіти щодо активності працівників, пройдених курсів, тестових результатів і рівня залученості в навчальний процес.

Система доступів для налаштування доступів для різних груп працівників – адміністратори можуть призначати ролі та рівні доступу до курсів залежно від посадових обов'язків або відділів, що дозволяє персоналізувати навчання для кожної групи.

Створення навчальних шляхів – для компаній доступна функція комбінування курсів у “навчальні шляхи”, що дозволяє створити структуровані програми навчання для різних груп співробітників. Це дає змогу формувати кар'єрні траєкторії, автоматизувати процес підвищення кваліфікації та контролювати послідовність проходження матеріалів.

Платформа Udemу [1] (рис. 1.1) вирізняється сучасним, мінімалістичним дизайном, орієнтованим на зручність користувачів. Вона має інтуїтивно зрозуміле розташування елементів, де основний акцент зроблений на курсах, що дозволяє легко знайти потрібний матеріал.

Інтерфейс адаптований до різних розмірів екранів, що забезпечує комфортну роботу як на ПК, так і на мобільних пристроях. Завдяки простій навігації, пошуку та фільтрації курсів користувачі можуть швидко отримувати доступ до контенту та інструментів, що робить платформу легкою у використанні, навіть для новачків.

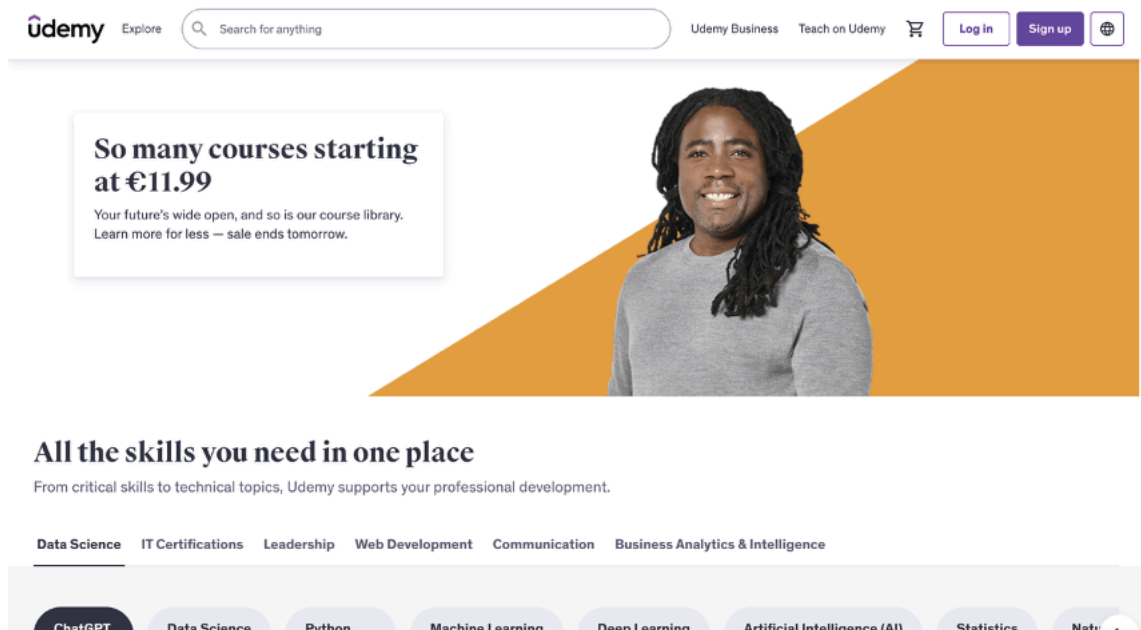


Рисунок 1.1 – Головна сторінка вебзастосунку платформи Udemy [1]

1.1.2 Аналіз платформи онлайн навчання: Coursera

Одна з найбільших онлайн платформ для навчання, Coursera [2] була заснована в 2012 році і надає доступ до курсів, спеціальностей, сертифікатів та дипломів, розроблених провідними університетами та компаніями. Платформа підтримує як безкоштовні, так і платні курси, даючи студентам можливість обирати оптимальний варіант для навчання.

Зростання популярності Coursera [2] стало можливим завдяки кільком ключовим факторам, включаючи партнерства з університетами та компаніями, а також високоякісному контенту, який пропонується на платформі. Одним із основних факторів, який відрізняє Coursera [2] від інших платформ, є її орієнтація на академічний підхід до навчання. Більшість курсів

створюються в партнерстві з університетами, такими як Stanford, Yale, University of London, що дозволяє пропонувати якісні академічні програми, які можна проходити онлайн. Це дозволяє студентам отримати знання від найкращих фахівців у своїй галузі та отримати сертифікати, що визнаються роботодавцями і академічними установами. Інноваційний підхід до навчання включає використання методів активного навчання, таких як проєктні роботи, завдання, інтерактивні відео та групові дискусії. Це дозволяє студентам не лише отримувати теоретичні знання, а й застосовувати їх на практиці.

Модель доступу до курсів на Coursera [2] є гнучкою, адже студенти можуть обирати між безкоштовним доступом до матеріалів курсів або оплатою за сертифікат, що підтверджує завершення курсу. Для більш серйозних студентів доступні спеціалізації та дипломні програми, які дозволяють отримати сертифікати, що визнаються роботодавцями. Coursera [2] також пропонує можливості для фінансової підтримки, наприклад, знижки або безкоштовні сертифікати для тих, хто має обмежений доступ до фінансів.

Проаналізувавши платформу Coursera [2], можна виокремити її унікальні функціональні можливості:

Функціональна група для студентів:

Спеціалізації та професійні сертифікати – студенти мають можливість отримати спеціалізації, розроблені провідними університетами та компаніями, такими як Google, IBM, Meta. Сертифікати можна додавати до профілів багатьох HR платформ таких як LinkedIn і вони визнаються багатьма роботодавцями.

Проєктні роботи під керівництвом – це унікальний спосіб навчання через практичні завдання в хмарному середовищі з покроковими інструкціями. Студенти працюють з актуальними інструментами та технологіями, отримуючи реальний досвід що забезпечує більш краще засвоєння інформації.

Система оцінки ровесниками – інноваційний метод оцінювання, де студенти оцінюють роботи своїх одногрупників за допомогою чітко визначених критеріїв, що сприяє глибшому розумінню матеріалу та розвитку критичного мислення.

Функціональна група для викладачів:

Студія створення контенту – спеціалізована платформа для створення професійного навчального контенту з інтегрованими інструментами для запису відео, створення інтерактивних завдань і візуалізації даних.

Аналітична панель для навчання – потужна панель аналітики, що використовує AI для аналізу звичок навчання студентів, прогнозування відсіву та оптимізації процесу навчання.

Автоматизована система перевірки коду – інструменти для автоматичної перевірки програмного коду з можливістю надання персоналізованих відгуків викладачів.

Корпоративне навчання для бізнесу:

Academy для бізнесу – спеціалізована програма для HR-менеджерів та фахівців з навчання та розвитку, яка надає інструменти для створення індивідуальних навчальних траєкторій і оцінки повернення інвестицій у навчання.

Технологія картографії навичок – інноваційна технологія, що дозволяє компаніям визначати прогалини в навичках співробітників і автоматично пропонувати відповідні курси для їх розвитку.

Платформа Coursera (рис. 1.2) вирізняється сучасним дизайном, орієнтованим на зручність користувачів, з інтуїтивно зрозумілою навігацією та широкими можливостями для фільтрації курсів. Платформа підтримує адаптивний дизайн для використання на різних пристроях, таких як комп'ютери, планшети та смартфони. Завдяки високоякісному контенту та партнерствам з провідними університетами, Coursera [2] залишається однією з найпопулярніших платформ для онлайн навчання у світі.

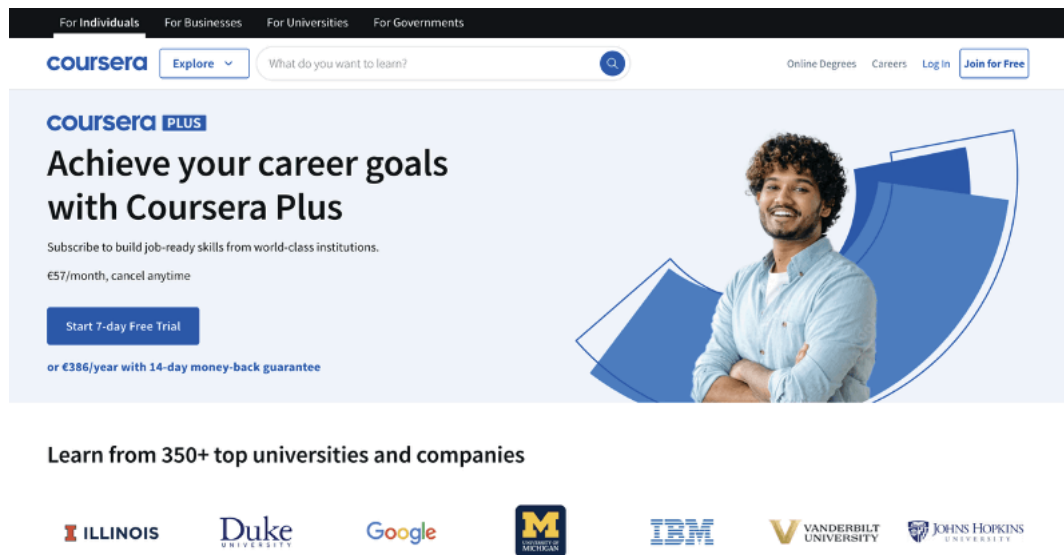


Рисунок 1.2 – Головна сторінка вебзастосунку платформи Coursera [2]

1.1.3 Аналіз платформи онлайн навчання: Educative

Educative [3] є онлайн платформою для навчання, яка надає можливість студентам проходити інтерактивні курси, що фокусуються на технічних та програмувальних навичках. Платформа, заснована у 2012 році, відрізняється від інших онлайн освітніх платформ тим, що пропонує учням можливість не тільки вивчати матеріали, але й практично застосовувати отримані знання під час проходження курсу. Educative [3] спеціалізується на створенні високоякісних інтерактивних програм, які дозволяють студентам виконувати код у реальному часі, не виходячи з платформи.

Ключовими факторами, що сприяли популярності платформи Educative [3] на початку її існування, був інтерактивний підхід до навчання, орієнтація на технічні навички та фокус на самотійному вивченні. Платформа надає можливість студентам отримувати навички програмування та інших технічних дисциплін безпосереднє виконання задач, що розвиває їх практичні навички й дає змогу глибше розуміти теоретичні концепти.

Особливості функціонування Educative [3] включають безперервне навчання через доступ до інтерактивних завдань, що дозволяє студентам

одразу застосовувати свої знання. Курси фокусується на конкретних технічних темах, таких як програмування, алгоритми, структури даних, розробка програмного забезпечення, розробка вебзастосунку та інші важливі навички для ІТ-спеціалістів.

Проаналізувавши платформу Educative [3], можна виокремити кілька груп функціональних можливостей, які виокремлюють її серед конкурентів.

Функціональна група для студентів:

Інтерактивні курси з практичними завданнями – унікальна особливість Educative полягає в тому, що всі курси інтерактивні, і студенти можуть одразу виконувати завдання на платформі. Це допомагає закріпити теоретичні знання через практичні вправи в реальному часі.

Модульна структура курсів – курси на Educative [3] розбиті на малі модулі, що дозволяє студентам вивчати нові концепти поетапно. Кожен модуль містить як теоретичні матеріали, так і практичні завдання для самостійного виконання.

Навчання без необхідності додаткових інсталяцій – усі курси на Educative [3] підтримують вбудовану інтеграцію з інтерфейсами для написання коду без необхідності використовувати додаткові інструменти чи середовища. Це спрощує навчальний процес для студентів.

Персоналізація навчання – Educative [3] дозволяє студентам налаштовувати власний темп навчання. Користувачі можуть вибирати курси, що найбільше відповідають їхнім інтересам і рівню підготовки, а також мають можливість проходити тести для перевірки своїх знань на кожному етапі.

Функціональна група для викладачів:

Інструменти для створення контенту – Educative [3] надає спеціальну платформу для викладачів, яка дозволяє створювати інтерактивні курси, що включають не лише текстові матеріали, але й кодові завдання, що інтегруються безпосередньо в процес навчання. Інструменти підтримують

різні формати контенту, включаючи відео, текстові матеріали та інтерфейси для кодування.

Аналітика студентів – викладачі мають доступ до статистики та аналітики, що дає змогу відстежувати успіхи студентів у виконанні завдань, а також здійснювати корекцію курсу на основі даних про прогрес учнів.

Гнучкість у створенні курсів – викладачі можуть гнучко налаштовувати складність завдань та тривалість курсу залежно від рівня знань студентів. Платформа надає багато варіантів для формування контенту, що дозволяє створювати персоналізовані навчальні програми.

Educative [3] (рис. 1.3) має мінімалістичний та інтуїтивно зрозумілий інтерфейс, який забезпечує зручну навігацію та простоту в користуванні. Платформа адаптована до різних типів пристроїв, включаючи десктопи, планшети та мобільні телефони, що гарантує комфортне навчання в будь-якому середовищі. Також підтримуються мобільні застосунки для iOS та Android, що дозволяє студентам продовжувати навчання на ходу. Прогрес синхронізується між різними пристроями, що забезпечує зручність та ефективність у навчальному процесі.

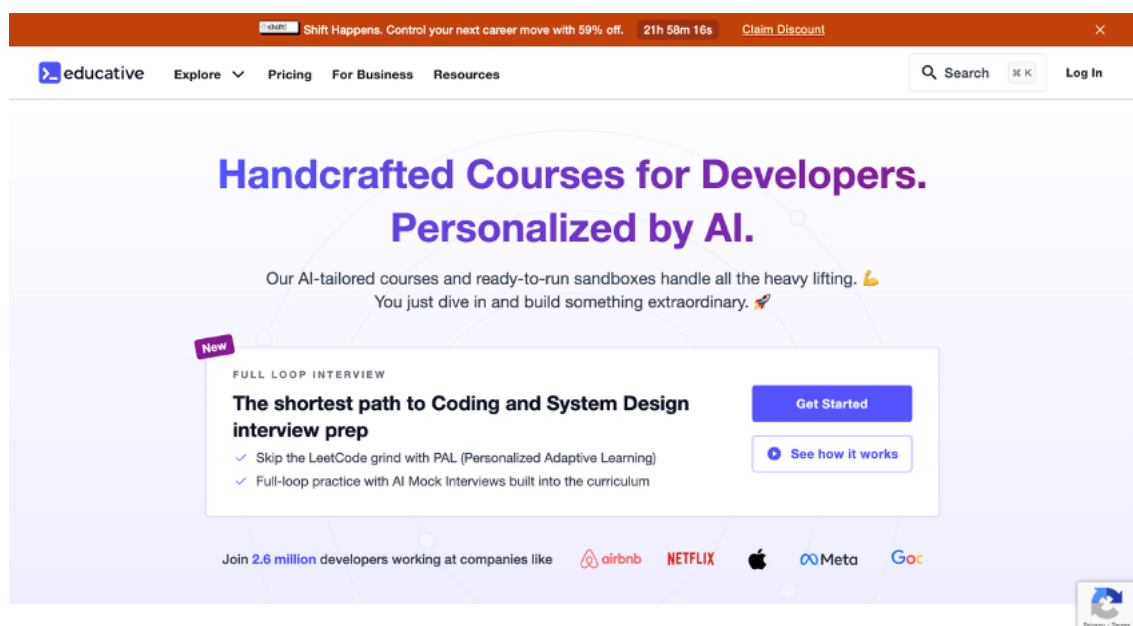


Рисунок 1.3 – Головна сторінка вебзастосунку платформи Educative

[3]

1.1.4 Висновки щодо існуючих вебплатформ онлайн навчання

Після аналізу наявних рішень було складено таблицю № 1.1, яка містить порівняння вебплатформ для онлайн навчання.

Таблиця 1.1 – Порівняльний аналіз вебплатформ для онлайн навчання.

Категорія	Udemy	Coursera	Educative
Формат навчання	Відеокурси	Відеокурси, інтерактивні завдання, сертифікати	Інтерактивні курси (кодинг у браузері, тести)
Тип контенту	Курси від незалежних інструкторів	Курси від університетів та компаній	Курси для програмістів та ІТ спеціалістів
Сертифікація	Немає офіційних сертифікатів	Фіційні сертифікати, можливість отримання дипломів	Підписка
Ціни	Разова оплата за курс (часті знижки)	Підписка або разова оплата, залежить від курсу	Сертифікація після завершення курсів
Якість контенту	Залежить від викладача	Висока, оскільки курси від університетів та компаній	Підписка
Кращий для ...	Самостійного навчання за низькою ціною	Академічного навчання, сертифікації	Інтерактивного навчання програмуванню

Продовження таблиці 1.1.

Категорія	Udemy	Coursera	Educative
Популярні категорії	Програмування, бізнес, дизайн, мовні курси	Академічні дисципліні, технічні навички, data science	Програмування, алгоритми, system design
Доступ до курсів	Назавжди після покупки	Обмежений час	Доступний поки активна підписка
Можливість практики	Залежить від курсу	Практичні завдання, проєкти	Інтерактивний кодинг безпосередньо в браузері

1.2 Постановка завдання

На перший погляд платформи Udemy [1], Coursera [2] та Educative [3] можуть здаватися схожими, але детальніший аналіз показує, що кожна з них орієнтована на різні групи користувачів.

Таким чином платформа Udemy [1] надає доступ до великої кількості курсів за помірну ціну. Це чудовий варіант для тих, хто хоче навчатися без великих фінансових витрат, проте сертифікати, які вона надає, зазвичай не визнаються роботодавцями як офіційне підтвердження навичок. Враховуючи ще можна сказати що головними користувачами платформи є студенти які бажають отримати знання в альтернативній формі або користувачі яким не важливе підтвердження сертифікату зазвичай такі люди фрілансери або індивідуальні підприємці.

Coursera [2] вирізняється тим, що пропонує не лише освітній контент, але й можливість отримати визнані сертифікати завдяки партнерству з провідними університетами та компаніями. Така особливість робить

платформу привабливою для фахівців, які прагнуть кар'єрного зростання на поточній посаді або планують змінити професійний напрямок. Офіційне підтвердження здобутих знань у вигляді сертифікату від авторитетних закладів значно посилює позиції спеціаліста на ринку праці.

Educative [3] в свою чергу орієнтована на тих, хто хоче активно практикуватися. Платформа пропонує інтерфейс для кодування, що дозволяє користувачам одразу застосовувати отримані знання на практиці. Це важливо для початківців, які хочуть розвивати практичні навички, саме тому основними користувачами платформи стють люди які намагаються здобути навички у новій сфері бо теоретична інформація яка була використана набагато краще запам'ятовується.

Проаналізувавши ці платформи, було створено діаграму за такими критеріями:

- складність курсу – критерій складності засвоєння інформації під час проходження курсу. Показує необхідний рівень знань користувача для проходження курсів платформи;
- якість курсу – критерій яких показує наскільки добре користувачі розуміють засвоєні навички після проходження курсу на платформі онлайн навчання;
- визнання сертифікатів роботодавцями – критерій визнання роботодавцями сертифікатів платформи онлайн навчання які користувачі отримують після проходження курсу;
- практичні можливості – критерій націленості платформи на практикування отриманих навичок користувача на практиці після проходження теоретичної частини;
- підтримка та взаємодія з користувачами – підтримка користувачів платформи у випадку виникнення проблеми у користувачів.

Результати аналізу платформи представлені у вигляді діаграми на рисунку 1.4.

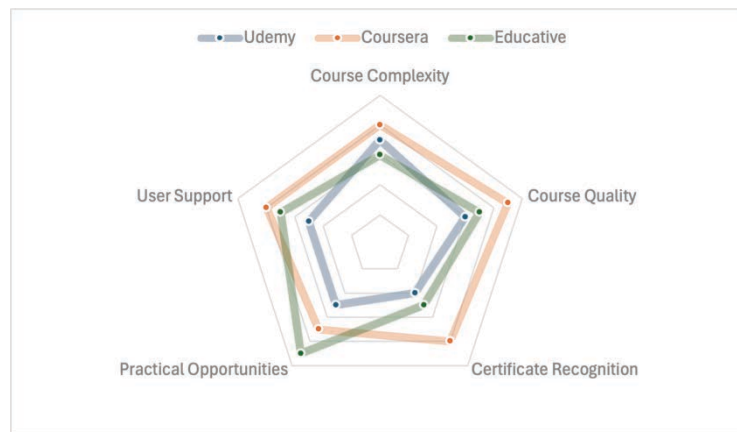


Рисунок 1.4 – Діаграма порівняння платформ UdeMy [1], Coursera [2] та Educative [3]

Складність курсів:

UdeMy [1] – різноманіття курсів з різними рівнями складності. Є курси для початківців, середнього рівня та більш просунутих користувачів.

Coursera [2] – курси часто мають академічний підхід і призначені для користувачів, які мають середній або високий рівень знань.

Educative [3] – платформа орієнтована на початківців та середній рівень. Курси містять багато практики, що допомагає вивчити матеріал на основі реальних задач.

Якість курсів:

UdeMy [1] – якість курсів може варіюватися, оскільки платформу можуть використовувати як професіонали, так і новачки. Багато курсів є доступними за рахунок масової пропозиції.

Coursera [2] – курси зазвичай створюються університетами та відомими компаніями, що забезпечує високу якість і професіоналізм.

Educative [3] – курси орієнтовані на високий рівень взаємодії з учнями, включають практичні завдання і створені для того, щоб користувачі могли на практиці застосовувати отримані знання.

Визнання сертифікатів роботодавцями:

UdeMy [1] – сертифікати не мають великої ваги на ринку праці, оскільки вони не завжди визнаються роботодавцями.

Coursera [2] – сертифікати визнаються багатьма роботодавцями та можуть бути використані для підтвердження кваліфікації.

Educative [3] – сертифікати не так широко визнаються на ринку праці, оскільки основний акцент робиться на практичні навички.

Практичні можливості:

Udemy [1] – в основному курси орієнтовані на теоретичне навчання, хоча є й деякі з практичними вправами.

Coursera [2] – більшість курсів включає відео, тести та інші методи навчання, але не завжди з фокусом на практику.

Educative [3] – велика увага приділяється практиці. Платформа пропонує реальні інтерактивні завдання, де користувачі можуть практикувати свої навички прямо в браузері.

Підтримка та взаємодія з користувачами:

Udemy [1] – платформа надає підтримку, але взаємодія між користувачами та інструкторами обмежена.

Coursera [2] – має більший рівень підтримки та можливості для взаємодії з викладачами, особливо в межах університетських програм.

Educative [3] – забезпечує активну підтримку та взаємодію з іншими студентами через коментарі та форуми, що сприяє більш ефективному навчальному процесу.

Проаналізувавши діаграму можна зробити наступні висновки. Сучасний ринок ІТ-технологій характеризується високим рівнем конкуренції та підвищеними вимогами до фахівців. Дослідження показують, що роботодавці при наймі працівників, особливо початківців, зосереджують увагу на практичних навичках кандидатів, надаючи другорядне значення наявності сертифікатів.

У цих умовах формується основна аудиторія навчальної платформи KnowWay.

Типовий користувач-студент – це амбітний молодий спеціаліст, який прагне отримати першу роботу в ІТ-сфері та потребує ґрунтовних

практичних знань. Для забезпечення якісної підготовки таких фахівців платформа залучає досвідчених викладачів-практиків.

Типовий користувач-викладач – це активний ІТ-спеціаліст із досвідом роботи від 3-5 років, який поєднує викладацьку діяльність з роботою в індустрії. Це може бути провідний розробник, технічний лід, архітектор програмного забезпечення або експерт у специфічних напрямках (DevOps, Data Science, Machine Learning, Cybersecurity, Cloud Computing).

Такий викладач має:

- актуальний досвід роботи в ІТ-компаніях;
- здатність ефективно передавати технічні знання;
- розуміння сучасних технологічних трендів;
- навички створення практичних завдань;
- прагнення розвивати молодих спеціалістів.

1.2.1 Функціональність системи

Визначивши портрети типових користувачів, можна сформулювати функціональні вимоги у вигляді історій користувачів.

Як студент, я хочу переглядати необхідні навички для популярних професій, щоб знати, які курси мені потрібні.

Як студент, я хочу зберігати потрібні мені курси до списку бажань.

Як студент, я хочу фільтрувати курси за мовою, ціною, темами та категоріями, щоб зручніше їх шукати.

Як студент, я хочу шукати курси за назвою, щоб швидко знаходити потрібні.

Як студент, я хочу сортувати знайдені курси за датою додавання (нові-старі, старі-нові) та за ціною (дешеві-дорогі, дорогі-дешеві) для зручного пошуку.

Як студент, я хочу зберігати свій прогрес у курсах, щоб легко повертатися до переглянутих уроків.

Як студент, я хочу завантажувати додаткові матеріали уроків для кращого засвоєння інформації.

Як студент, я хочу мати можливість скидати прогрес у курсах, щоб пройти їх заново.

Як студент, я хочу переміщати курси в архів, щоб приховати зайві.

Як студент, я хочу відновлювати курси з архіву, щоб мати до них доступ.

Як студент, я хочу, щоб курси автоматично переміщувалися в архів після їх завершення.

Як студент, я хочу купувати курси, щоб отримати до них доступ.

Як викладач, я хочу створювати власні курси.

Як викладач, я хочу переглядати аналітику та метрики курсів, щоб покращувати навчальні матеріали.

Як викладач, я хочу переглядати аналітику та метрики продажів, щоб збільшити свій заробіток.

Як викладач, я хочу приховувати курси з пошуку, щоб керувати їх доступністю.

Як викладач, я хочу видаляти курси, щоб управляти своїм контентом.

Як викладач, я хочу редагувати курси, щоб оновлювати або покращувати їх.

1.2.2 Вимоги до інтерфейсу

За для простоти орієнтування та зосередження користувачів платформи онлайн навчання на основних функціях плаформи було обрано простий дизайн у мінімалістичному стилі без зайвих елементів що зменшує навантаження на користувача а також покращує зручність використання та ефективність взаємодії.

Згідно з дослідженням [4], кольорова гамма платформи відіграє важливу роль у сприйнятті продукту. Саме тому для кольорової гамми

платформи було обрано відтінки блакитного які асоціюються з наукою та професіоналізмом, що сприяє формуванню позитивного враження про платформу.

Також важливо реалізувати адаптивність інтерфейсу, щоб він коректно відображався на різних пристроях, а саме: комп'ютерах, планшетах і смартфонах. Це дозволить користувачам комфортно працювати із системою незалежно від пристрою.

1.2.3 Нефункціональні вимоги до продукту

Одна з ключових нефункціональних вимог, високий рівень захисту даних користувачів, адже система обробляє конфіденційну інформацію, зокрема платіжні дані. Платформа повинна відповідати сучасним стандартам інформаційної безпеки, забезпечуючи надійний захист платежів і дотримання політики конфіденційності. Для цього необхідно використовувати шифрування даних, двофакторну аутентифікацію, а також захист від атак SQL Injection, XSS і CSRF.

Стабільність і безперебійна робота системи є критично важливими, оскільки користувачі оплачують доступ до сервісу. Будь-які збої можуть призвести до втрати довіри з боку користувачів, а також до фінансових збитків.

1.3 Висновки за розділом №1

У процесі аналізу існуючих освітніх платформ було визначено основні категорії користувачів, які становлять цільову аудиторію системи. Це, у свою чергу, дозволило сформулювати ключові вимоги до функціональних та нефункціональних можливостей платформи.

На основі отриманих вимог було створено діаграму (рис. 1.5), що демонструє переваги платформи KnowWay.

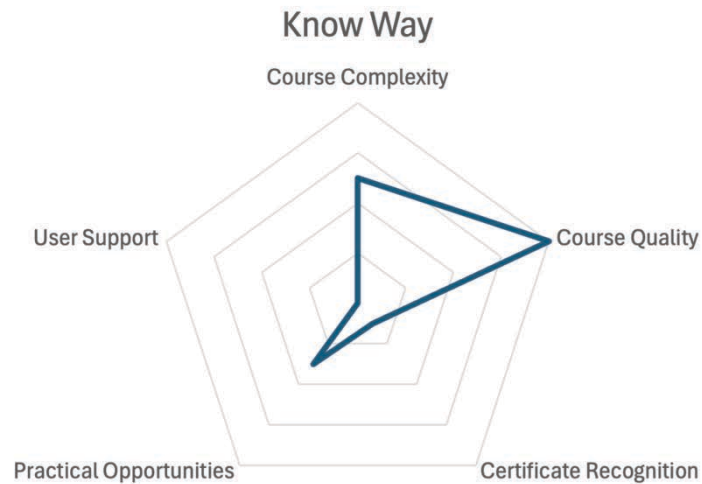


Рисунок 1.5 – Діаграма порівняння платформи KnowWay.

Зважаючи на обмежені ресурси на першочерговому етапі розробки, було прийнято рішення відмовитися від інтеграції системи підтримки користувачів. Однак, ця функціональність залишається важливим елементом платформи тому буде реалізована в наступних версіях платформи.

Крім того, подальший розвиток системи може включати розширення можливостей конструктора курсів для викладачів, зокрема, додавання вбудованого редактора коду. Це дозволить студентам практикувати набуті навички безпосередньо в системі, що підвищить залученість користувачів і зробить платформу більш привабливою для широкого кола аудиторії.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

Проаналізувавши усі доступні мови програмування для веброзробки програмного забезпечення а саме: Python, Ruby, PHP, JavaScript [6], Rust, GoLang, було обрано мову програмування TypeScript [5].

TypeScript [5] – це мова програмування, заснована на JavaScript [6], яка розширює його можливості, додаючи статичну типізацію, розширений набір синтаксичних конструкцій та покращену підтримку сучасних парадигм програмування. TypeScript [5] компілюється у стандартний JavaScript [6], що робить його сумісним із будь-яким середовищем, яке підтримує JavaScript [6].

Важливою перевагою мови програмування TypeScript [5] є можливість використання як для реалізації серверної частини програмного забезпечення так і можливість використання для реалізації клієнт частини програмного забезпечення.

Основними перевагами TypeScript [5] є:

- статична типізація, допомагає виявляти помилки на етапі розробки, що зменшує ризики появи багів у продакшені;
- сумісність із JavaScript [6], дозволяє поступово впроваджувати TypeScript у проекти, що вже використовують JavaScript [6];
- підтримка об'єктно-орієнтованого та функціонального програмування, зручні класи, інтерфейси та функціональні конструкції;
- широке використання в індустрії, багато популярних фреймворків (Angular [7], NestJs [8], NextJs [9]) побудовані на TypeScript [5] або мають глибоку інтеграцію з ним.

Вибір мови програмування TypeScript [5] зумовлено її популярністю та широким використанням у сфері веброзробки. Згідно з результатами опитування [10] (рис. 2.1), TypeScript [5] є однією з найпопулярніших мов серед розробників, що підтверджує його актуальність та ефективність у

використанні мови програмування у процесі створення сучасного програмного забезпечення.

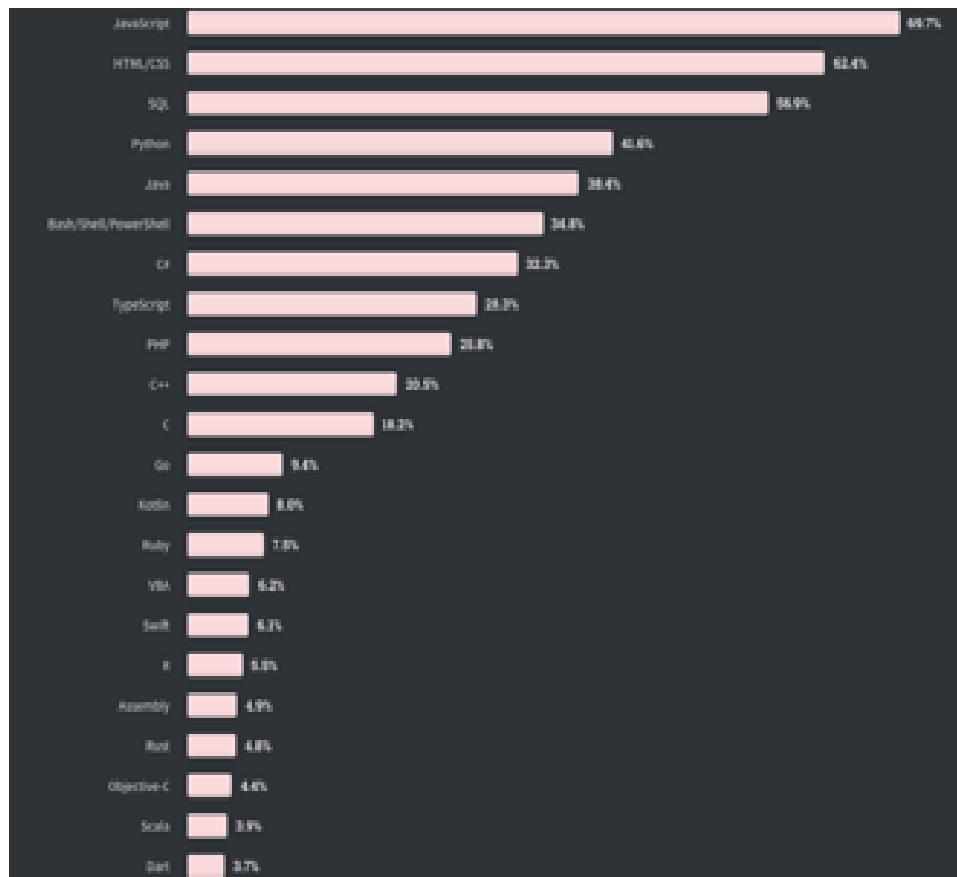


Рисунок 2.1 – Результати дослідження стосовно популярності використання мов програмування [11]

2.2 Вибір середовища розробки

Для створення програмного забезпечення на мові TypeScript [5] було обрано середовище розробки Visual Studio Code [12].

Visual Studio Code [12] є одним із найпопулярніших текстових редакторів та середовищ розробки, яке широко використовується розробниками у сфері веброзробки завдяки його потужним можливостям, розширюваності та підтримці великої кількості мов програмування.

Основними перевагами Visual Studio Code [12] є:

Легка вага та висока продуктивність, швидкий запуск та оптимізована робота навіть на слабких пристроях.

Широка екосистема розширень, підтримка численних плагінів, які дозволяють інтегрувати додатковий функціонал.

Вбудована підтримка Git [13], зручна система контролю версій без необхідності використання додаткового ПЗ.

Інтелектуальне автодоповнення коду, покращена продуктивність завдяки IntelliSense [14].

Гнучкість налаштувань, можливість зміни інтерфейсу, створення власних конфігурацій та кастомізація функціоналу.

У порівнянні з іншими IDE, такими як WebStorm [15], Atom [16], Sublime Text [17] та Eclipse [18], Visual Studio Code [12] вирізняється оптимальним балансом між функціональністю, швидкістю роботи та доступністю. WebStorm [15] має розширену підтримку JavaScript [6] і TypeScript [5], але є платним хоч і має безоплатну підписку для студентів потрібен час на підтвердження статусу студента. Atom [16] і Sublime Text [17] є легшими текстовими редакторами, проте поступаються VS Code у можливостях. Eclipse [18] більше орієнтований на Java [19], а підтримка TypeScript [5] у ньому є обмеженою. Завдяки своїм перевагам Visual Studio Code [12] став оптимальним вибором для розробки програмного забезпечення на TypeScript [5].

2.3 Вибір фреймворку

2.3.1 Вибір фреймворку для клієнтської частини

Проаналізувавши існуючі рішення для розробки клієнтської частини, за результатами дослідження [20] для аналізу було обрано п'ять найпопулярніших фреймворків: NextJs [9], NuxtJs [21], Angular [7], jQuery [22], Svelte [23]. Порівняння фреймворків для розробки клієнтської частини вебзастосунків наведено у таблиці 2.1.

Таблиці 2.1 – Порівняння фреймворків для клієнтської частини.

Фреймворк	Переваги	Недоліки
NextJs	Серверний рендеринг (SSR), статична генерація (SSG), чудова SEO-оптимізація, продуктивність, проста інтеграція з TypeScript.	Вимагає певного рівня знань React [24].
NuxtJs	Аналог Next.js для Vue.js, підтримує SSR та SSG, зручний для Vue-розробників.	Менша популярність у порівнянні з React, складніший у налаштуванні.
Angular	Повноцінний фреймворк з великою екосистемою, підтримка TypeScript, двостороннє зв'язування даних.	Високий поріг входу, складна структура.
jQuery	Простота у використанні, велика кількість плагінів.	Застарілий підхід, низька продуктивність для сучасних SPA-застосунків.
Svelte	Висока продуктивність, компілюється у ванільний JavaScript, простий у використанні.	Менша екосистема та підтримка у порівнянні з React або Vue.

З огляду на переваги та популярність, для розробки клієнтської частини було обрано NextJs [9], оскільки він забезпечує чудову продуктивність, підтримує серверний рендеринг і добре інтегрується з TypeScript [5].

2.3.2 Вибір фреймворку для серверної частини

Проаналізувавши існуючі рішення для розробки серверної частини, за результатами дослідження [34] для аналізу було обрано два найпопулярніші фреймворки: ExpressJs [25] і NestJs [8]. Порівняння фреймворків для розробки серверної частини вебзастосунку наведено у таблиці 2.2.

Таблиця 2.2 – Порівняння фреймворків для серверної частини

Фреймворк	Переваги	Недоліки
NestJS	Модульна архітектура, підтримка TypeScript, вбудована підтримка декораторів, добре підходить для великих проєктів.	Вища складність у порівнянні з Express.js.
Express.js	Простота у використанні, легка вага, велика спільнота.	Відсутність структурованої архітектури, необхідність додаткових бібліотек для масштабних застосунків.

Зважаючи на модульність, гнучкість та інтеграцію з TypeScript [5], для серверної частини було обрано NestJs [8], оскільки він забезпечує кращу архітектуру та підтримку сучасних підходів до розробки.

2.4 Вибір бази даних та ORM

Проаналізувавши усі існуючі ORM-системи та бази даних для мови програмування TypeScript [5]. Було обрано TypeORM [27] та базу даних PostgreSQL [26]. Порівняння ORM-систем наведено у таблиці 2.3.

Таблиця 2.3 – Порівняння ORM-систем

ORM-система	Переваги	Недоліки
TypeORM	Підтримка декораторів, інтеграція з TypeScript, гнучкість у налаштуванні, підтримка різних БД.	Дещо складніша у налаштуванні порівняно з іншими ORM.
Sequelize	Велика спільнота, підтримка різних БД, простий у використанні.	Відсутність повної підтримки TypeScript, низька продуктивність у складних запитах.
Prisma	Зручний API, автоматична генерація типів, висока продуктивність.	Обмежена підтримка складних запитів, менш гнучка конфігурація.

Зважаючи на повну підтримку TypeScript [5], гнучкість та масштабованість, було обрано TypeORM [27].

2.5 Розробка моделі взаємодії користувача

У процесі розробки програмного забезпечення було створено модель взаємодії користувачів із системою, яка відображає основні сценарії використання платформи користувачем. Система надає можливості як для звичайних користувачів, так і для викладачів, забезпечуючи ефективну взаємодію з курсами та іншими функціями платформи.

Кожен користувач платформи має можливість проходити процедуру автентифікації, створюючи новий обліковий запис або входячи до вже існуючого. Після авторизації користувач може переглядати список курсів

придбаних курсів, а також отримувати персоналізовані рекомендації щодо нових курсів.

Система надає доступ до каталогу всіх доступних курсів, у якому користувач може здійснювати пошук за категоріями, популярністю або рівнем складності. Крім того, користувач може купувати нові курси, отримувати доступ до їхнього змісту, переглядати навчальні матеріали та проходити уроки. За необхідності він може переміщувати завершені або неактуальні курси до архіву для зручного керування навчальним процесом.

Користувачі, які мають статус викладача, отримують розширений набір можливостей. Вони можуть створювати нові навчальні курси для інших користувачів, редагувати їхній зміст, додавати та оновлювати уроки. Крім того, система надає їм інструменти для управління контентом, зокрема можливість змінювати порядок занять, додавати тести та завдання для закріплення матеріалу.

Викладачі також мають доступ до аналітики, що дозволяє переглядати активність студентів на курсах, аналізувати їхній прогрес і отримувати статистику щодо перегляду матеріалів.

Основний успішний сценарій взаємодії користувача з вебзастосунком можна описати таким чином:

- користувач проходить автентифікацію в системі, створюючи новий обліковий запис або входячи до наявного;
- після входу система відображає інформацію про активні курси, які користувач проходить на даний момент, а також пропонує рекомендовані курси на основі його інтересів;
- користувач здійснює навігацію по застосунку: переглядає уроки, додає курси до архіву, купує нові навчальні програми, здійснює пошук у каталозі;
- після завершення роботи користувач виходить із системи або просто закриває застосунок.

Для кращого розуміння логіки роботи користувачів із системою модель взаємодії представлена у вигляді діаграми прецедентів (рис. 2.2).

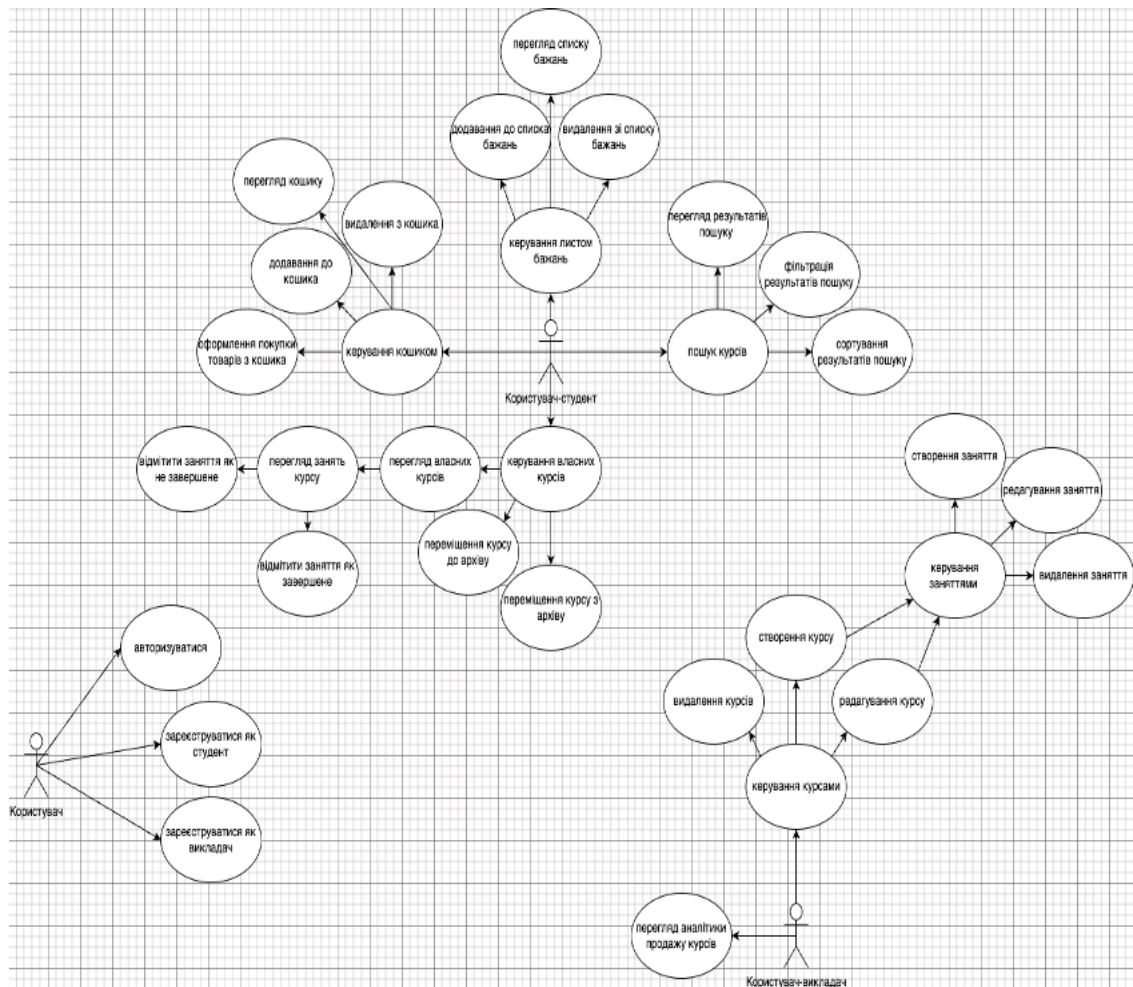


Рисунок 2.2 – Модель взаємодії користувача з програмним забезпеченням вебзастосунку для проходження онлайн курсів

Ця діаграма наочно відображає всі можливі дії користувачів та їхній взаємозв'язок із функціональними можливостями системи.

2.6 Висновки за розділом 2

У даному розділі було обґрунтовано вибір технологій для розробки програмного забезпечення. Було визначено, що TypeScript [5] є оптимальною мовою програмування завдяки статичній типізації, сумісності з JavaScript [6],

підтримці сучасних парадигм програмування та його широкому використанню у веброзробці.

Для розробки програмного забезпечення було обрано Visual Studio Code [12] як основне середовище розробки через його популярність, широку екосистему розширень, довану підтримку Git [13] та інтелектуальне автодоповнення коду.

При виборі фреймворків для клієнтської та серверної частин було проведено аналіз популярних рішень. Для розробки клієнтської частини вебзастосунку було обрано NextJs [9], оскільки він забезпечує високу продуктивність, підтримку серверного рендерингу та ефективну інтеграцію з TypeScript [5]. Для серверної частини було обрано NestJs [8], який відзначається модульною архітектурою, гнучкістю та підтримкою TypeScript [5].

Вибір системи управління базами даних здійснювався з урахуванням продуктивності, підтримки TypeScript [5] та можливостей масштабування. У результаті було обрано PostgreSQL [26] у поєднанні з TypeORM [27] як ORM-систему, що забезпечує гнучкість, підтримку декораторів та інтеграцію з TypeScript [5].

Також у процесі розробки було створено модель взаємодії користувачів із системою, яка забезпечує ефективну роботу як звичайних користувачів, так і викладачів, дозволяючи їм зручно взаємодіяти з навчальними курсами та іншими функціями платформи.

3 ОПИС ТА РЕАЛІЗАЦІЯ ПРОГРАМИ

3.1 Вимоги до технічних і програмних засобів

Вимоги до технічних та програмних засобів визначають мінімальні характеристики, які повинні мати клієнтські пристрої для коректної роботи вебзастосунку.

Вимоги:

- процесор Intel Core i3-8100F з тактовою частотою, ГГц 3,6 або еквівалентний;
- оперативна пам'ять: DDR3 з об'ємом мінімум 4 Гб;
- засоби виводу інформації: монітор з роздільною здатністю не менше 1280x800 пікселів;
- дисковий простір: мінімум 1 Гб вільного місця на жорсткому диску.

Вимоги до програмного забезпечення:

- веббраузер: останні версії Google Chrome, Mozilla Firefox, Microsoft Edge, Safari або аналогічних;
- інтернет-з'єднання: Стабільне з'єднання з Інтернетом для завантаження та використання вебзастосунку;
- підтримка JavaScript: увімкнена підтримка веббраузером для виконання скриптів JavaScript;
- операційна система: Windows 10 або вище, macOS 10.13 або вище, Linux (Ubuntu, Fedora тощо).

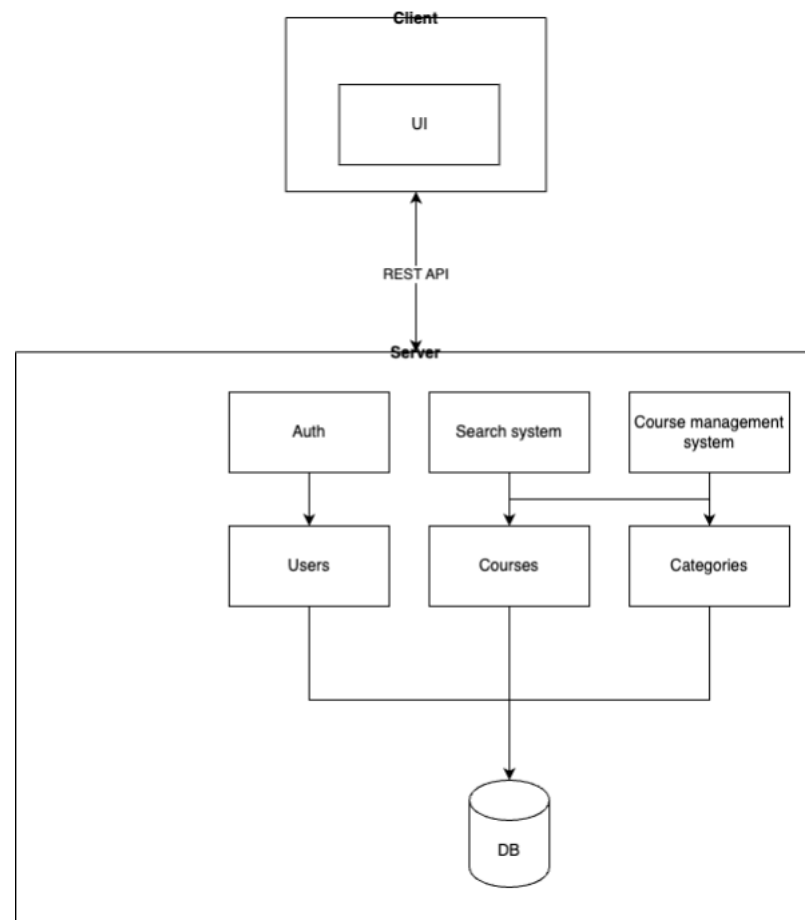
3.2 Архітектура програмного забезпечення

Планування архітектури програмного забезпечення це найважливіший етап у розробці програмного забезпечення бо саме цей фактор найбільш впливає на швидкість розробки, бо добре спроектована система дозволить

зручно реалізувати систему та запобігти незапланованого переписування коду.

Архітектура програми визначає структуру, компоненти та взаємодію між ними, що визначає робочі процеси, ефективність та масштабованість застосунку.

Для реалізації програмного забезпечення вебзастосунку онлайн курсів було використано стандартну клієнт серверну архітектуру (рис. 3.1).



Рисуно 3.1 – Діаграма архітектури вебзастосунку

Як можна побачити з діаграми система складається з двох частин, клієнтська частина вебзастосунку відповідає за відображення інтерфейсу користувача та серверної частини яка в свою чегру теж поділяється на дві частини.

Перша частина відповідає за функціонал пов'язаний з студентами викладачами та друга частина яка відповідає за функціонал пов'язаний з користувачами студентами.

3.3 Розробка модулів системи

Система поділена на модулі які взаємодіють один з одним. Кожен модуль виконує окрему функцію, що дозволяє легко розширювати систему та інтегрувати нові можливості без впливу на інші компоненти.

Модуль Auth відповідає за авторизацію та реєстрацію користувачів, управління сесіями та відновлення паролів. Він забезпечує аутентифікацію користувачів за допомогою логіна та пароля. Реалізовано функціонал реєстрації нових користувачів, відновлення доступу (скидання пароля). Додатково передбачено керування сесіями, включаючи їх завершення та оновлення токенів.

Модуль User призначений для керування обліковими записами користувачів. Він надає можливість редагування профілю, зміни пароля, оновлення аватара та контактної інформації. Адміністратори мають змогу переглядати список користувачів, призначати ролі, налаштовувати права доступу, а також блокувати або видаляти акаунти.

Модуль Course забезпечує створення, редагування, видалення та перегляд курсів. Він дозволяє адміністраторам або викладачам додавати нові курси, наповнювати їх навчальними матеріалами, змінювати зміст, публікувати або приховувати курси від користувачів. Передбачено можливість категоризації курсів, встановлення рівнів складності та налаштування доступу.

Модуль Lesson відповідає за керування заняттями курсів. Кожен урок може містити різні навчальні матеріали, такі як відео, текстові документи, інтерактивні завдання та тести. Модуль підтримує функціонал відстеження

прогресу студентів у навчанні, а також взаємодію з матеріалами (наприклад, коментування або позначення завершених уроків).

Модуль Category забезпечує управління категоріями курсів. Він дозволяє адміністраторам додавати нові категорії, змінювати їхні назви та ієрархію, а також видаляти непотрібні категорії. Це сприяє впорядкуванню курсів та покращує навігацію в системі.

3.4 Розробка алгоритмів

3.4.1 Алгоритм авторизації

Процес авторизації користувача (рис. 3.3) забезпечує можливість входу в систему після успішної авторизації.

Спочатку користувач проходить процедуру реєстрації (рис. 3.2), вводячи електронну пошту та пароль.

Введені дані проходять валідацію, що включає перевірку формату електронної пошти, відповідності пароля мінімальним вимогам безпеки та унікальності електронної пошти в базі даних.

Якщо всі перевірки успішні, система створює обліковий запис і надсилає на вказану адресу електронний лист із посиланням для підтвердження реєстрації. Після активації облікового запису користувач отримує можливість входу в систему.

Авторизація передбачає введення електронної пошти та пароля у відповідні поля форми входу.

Система перевіряє наявність облікового запису, порівнює введений пароль із хешованим значенням, що зберігається в базі даних, і в разі успішного збігу створює сеансовий токен або встановлює сесію.

У відповідь користувач отримує підтвердження успішної авторизації та доступ до функціоналу системи відповідно до своїх прав.

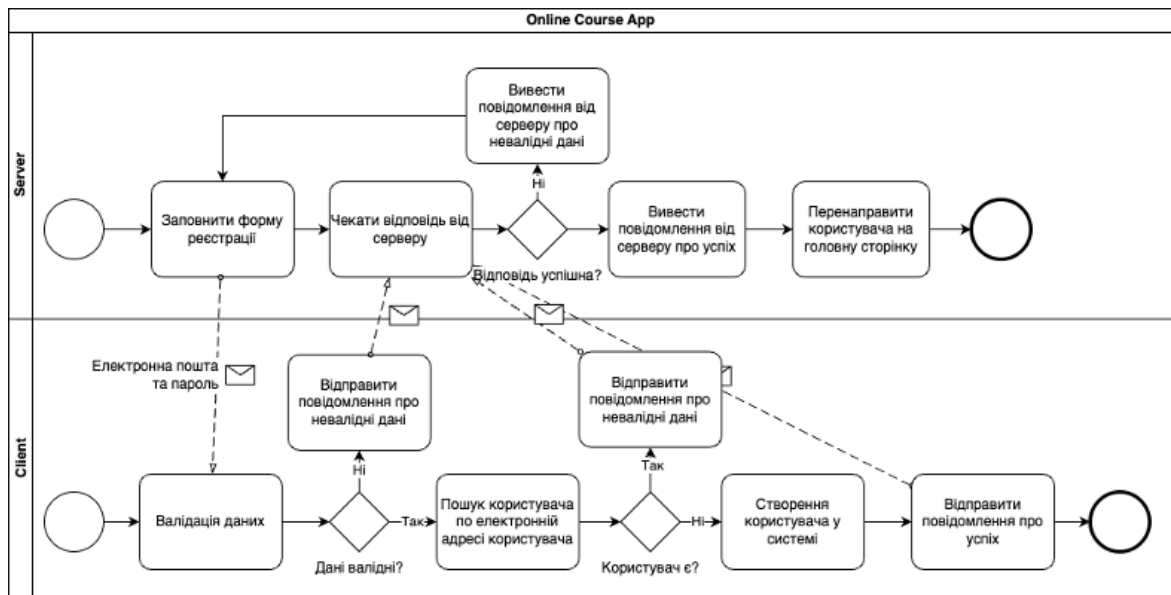


Рисунок 3.2 – Алгоритм реєстрації користувача

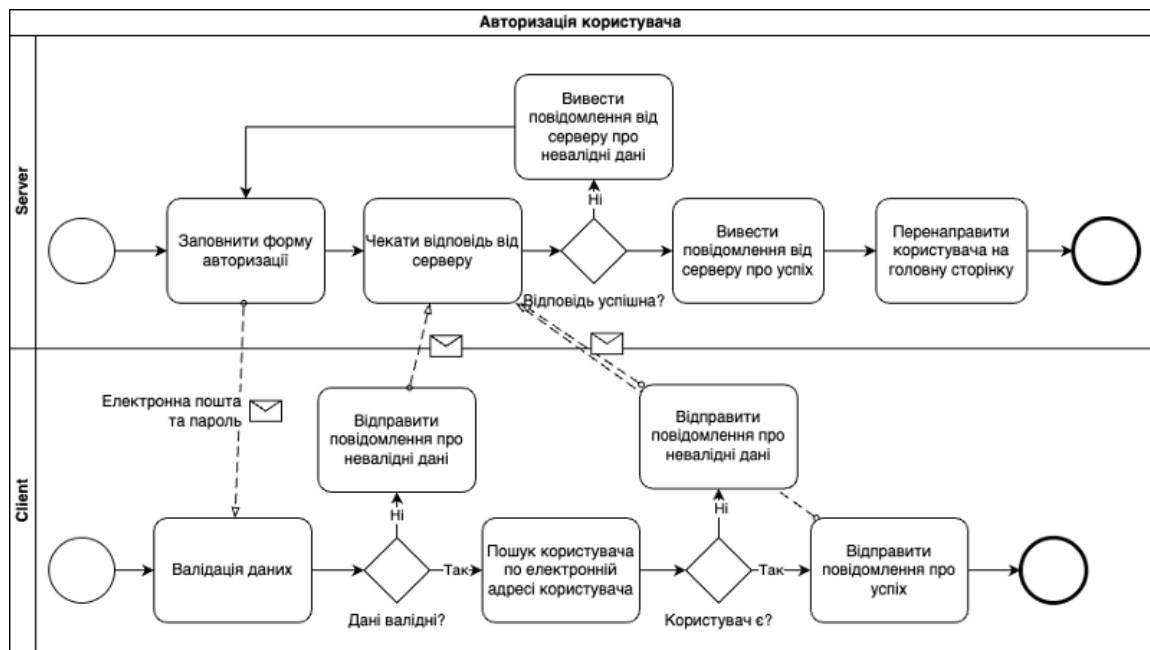


Рисунок 3.3 – Алгоритм авторизації користувача

3.4.2 Алгоритм оновлення даних користувача

Авторизований користувач може оновити персональні дані (рис. 3.4) у своєму особистому кабінеті.

Для цього потрібно натиснути на аватар користувача та вибрати пункт меню “Налаштування профілю”.

У цьому розділі передбачено можливість редагування даних, згрупованих за окремими категоріями. У групі “Безпека” користувач може змінити пароль, а в групі “Основне” – оновити ім’я та прізвище.

Крім того, у налаштуваннях доступна опція виходу з облікового запису. Кожна група має окремий розділ, що забезпечує зручну навігацію та швидкий доступ до необхідних параметрів.

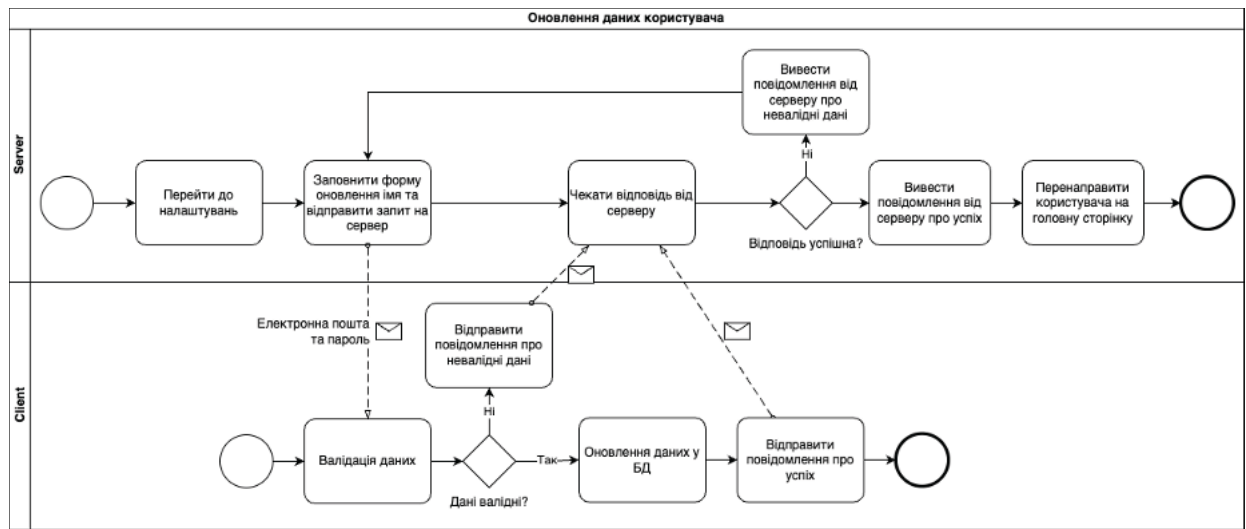


Рисунок 3.4 – Алгоритм оновлення персональних даних користувача

3.4.3 Алгоритм пошуку та фільтрації курсів

Користувач може шукати курси за назвою (рис. 3.5), фільтрувати їх за ціною та категорією, а також сортувати результати за датою створення або популярністю.

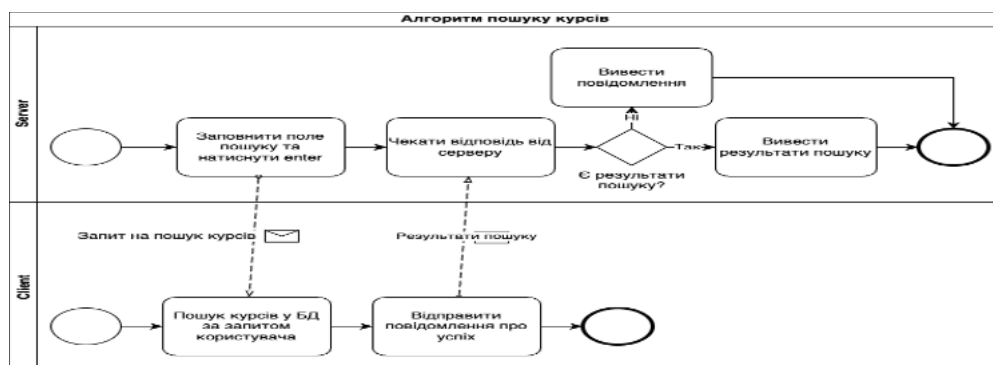


Рисунок 3.5 – Алгоритм пошуку та фільтрації курсів

3.4 Проектування бази даних

Одним з наступних етапів розробки програмного забезпечення є проектування структури бази даних, оскільки правильна структура даних забезпечує ефективність, масштабованість та безпеку системи.

У цьому проєкті для роботи з базою даних було обрано PostgreSQL [26], а для взаємодії з нею використовується ORM-інструмент TypeORM [27]. ER-діаграму проєкту наведено на рисунку 3.6.

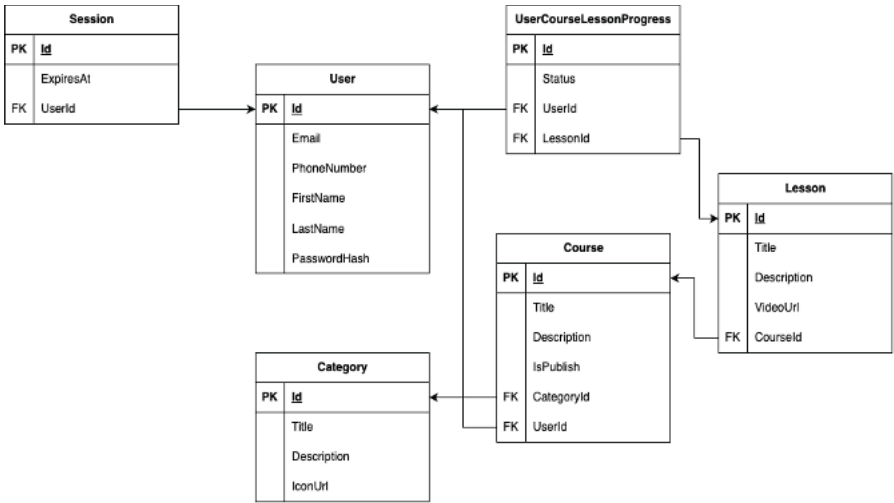


Рисунок 3.6 – ER-діаграма проєкту

Сутність «User» відображає дані про кожного зареєстрованого користувача. Опис схеми наведено у таблиці 3.1.

Таблиця 3.1 – Схема сутності «User»

Назва	Тип	Опис
Id	String	Унікальний номер користувача
FirstName	String	Ім'я користувача
LastName	String	Прізвище користувача
PasswordHash	String	Захешований пароль

Продовження таблиці 3.1

Назва	Тип	Опис
Role	“student” “teacher”	Роль користувача (студент або викладач)
Email	String	Електронна адреса користувача, використовується для авторизації

Сутність «Session» відображає дані про поточні сесії користувачів. Опис схеми наведено у таблиці 3.2.

Таблиця 3.2 – Схема сутності «Session»

Назва	Тип	Опис
Id	String	Унікальний номер сесії
UserId	String	Унікальний номер користувача
ExpiresAt	Date	Дата закінчення строку сесії

Сутність «Course» відображає дані про курси у системі. Опис схеми наведено у таблиці 3.3.

Таблиця 3.3 – Схема сутності «Course»

Назва	Тип	Опис
Id	String	Унікальний номер курсу
Title	String	Назва курсу
Description	String	Опис курсу

Продовження таблиці 3.3

Назва	Тип	Опис
ImageUrl	String	Посилання на зображення курсу
UserId	String	Унікальний номер автора курсу
Price	Number	Ціна курсу

Сутність «Lesson» відображає дані про уроки курсів. Опис схеми наведено у таблиці 3.4.

Таблиця 3.4 – Схема сутності «Lesson»

Назва	Тип	Опис
Id	String	Унікальний номер заняття
CourseId	String	Унікальний номер курсу
VideoUrl	String	Посилання на відео
Description	String	Текст заняття

Сутність «UserCourseLessonProgress» відображає прогрес користувачів у курсах. Опис схеми наведено у таблиці 3.5.

Таблиця 3.5 – Схема сутності «UserCourseLessonProgress»

Назва	Тип	Опис
Id	String	Унікальний номер сутності прогресу користувача
LessonId	String	Унікальний номер заняття

Продовження таблиці 3.5

Назва	Тип	Опис
Status	“in-progress” “done”	Статус прогресу
UserId	String	Унікальний номер користувача

3.5 Розробка користувацького інтерфейсу

Розробка користувацького інтерфейсу є важливим етапом оскільки саме інтерфейс користувача забезпечує зручність використання, інтуїтивно зрозумілу навігацію та позитивний досвід для користувачів. Він не лише відображає функціональні можливості програми, але й створює перше враження, яке часто стає вирішальним у виборі програмного продукту.

Процес розробки користувацького інтерфейсу поділяється на декілька етапів. Послідовність процесів зображена на рисунку 3.7.

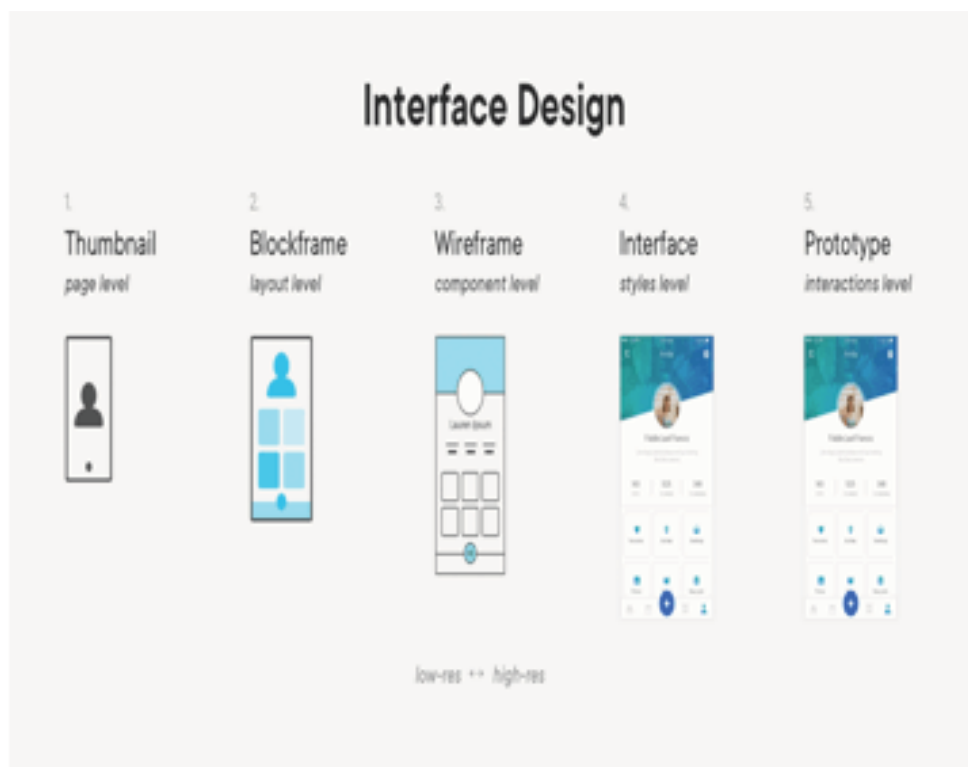


Рисунок 3.7 – Етапи розробки користувацького інтерфейсу

3.5.1 Розробка шаблонів

На цьому етапі створюються попередні макети екранів, які демонструють розташування основних елементів інтерфейсу без детального опрацювання їхнього дизайну. Ці макети слугують базою для подальшого доопрацювання візуальної складової та функціональності.

Для відображення логіки взаємодії користувача із системою використовується схема Wireframe Flow. Вона поєднує елементи wireframe з потоками переходів між екранами, що дає змогу оцінити не лише розташування елементів, а й сценарії навігації та послідовність дій користувача. Приклад Wireframe Flow представлений на рисунку 3.8.

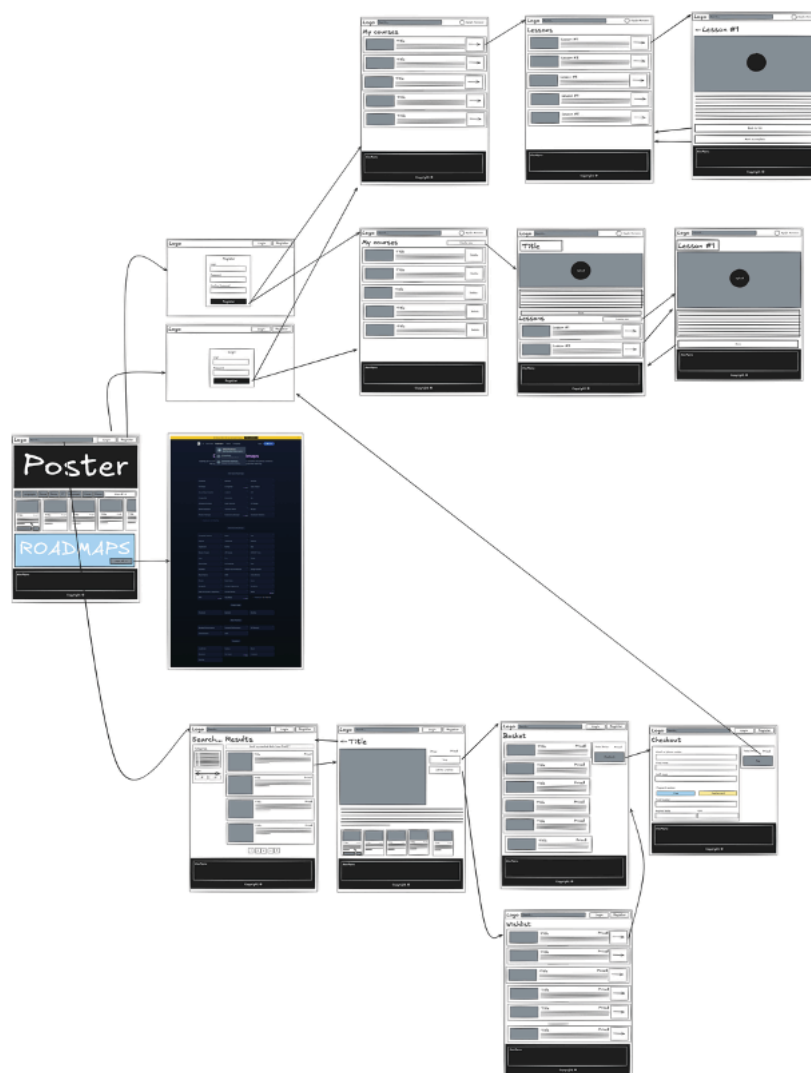


Рисунок 3.8 – Діаграма користувацького потоку

Цей підхід дозволяє краще зрозуміти структуру інтерфейсу ще на ранніх етапах розробки та виявити потенційні проблеми у користувацькому досвіді до створення детального дизайну.

Головна сторінка (рис. 3.9) має наступні елементи:

- хедер, поле для пошуку курсів та кнопки для реєстрації або авторизації;
- постер;
- каталог найпопулярніших курсів за категоріями;
- посилання на сайт з шляхами скілів;
- футер, меню навігації.

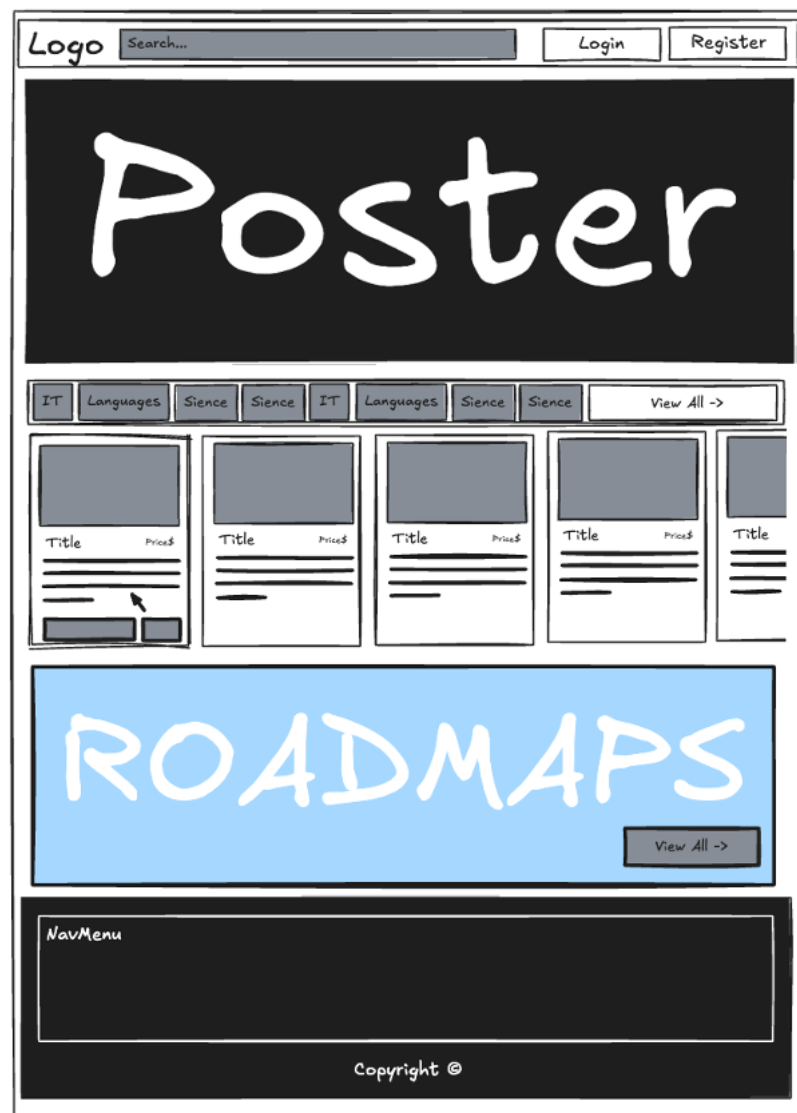


Рисунок 3.9 – Шаблон головної сторінки системи

Сторінки входу (рис. 3.10) та реєстрації користувачів (рис. 3.11), мають форму для входу або реєстрації, хедер та кнопки навігації

The wireframe shows a registration page template. At the top, there is a header bar containing a 'Logo' on the left and two buttons, 'Login' and 'Register', on the right. The main content area features a central registration form. This form is titled 'Register' and contains three input fields labeled 'Login', 'Password', and 'Confirm Password'. Below these fields is a dark 'Register' button.

Рисунок 3.10 – Шаблон сторінки реєстрації

The wireframe shows a login page template. At the top, there is a header bar containing a 'Logo' on the left and two buttons, 'Login' and 'Register', on the right. The main content area features a central login form. This form is titled 'Login' and contains two input fields labeled 'Login' and 'Password'. To the right of the 'Password' field is a link labeled 'Forgot password?'. Below these fields is a dark 'Register' button.

Рисунок 3.11 – Шаблон сторінки авторизації

Сторінка пошуку та фільтрації курсів (рис. 3.12) має такі елементи:

- хедер, поле для пошуку курсів та кнопки для реєстрації або авторизації;
- блок фільтрів;
- блок сортування результатів пошуку;
- блок результатів пошуку;
- пагінація;
- футер, меню навігації.

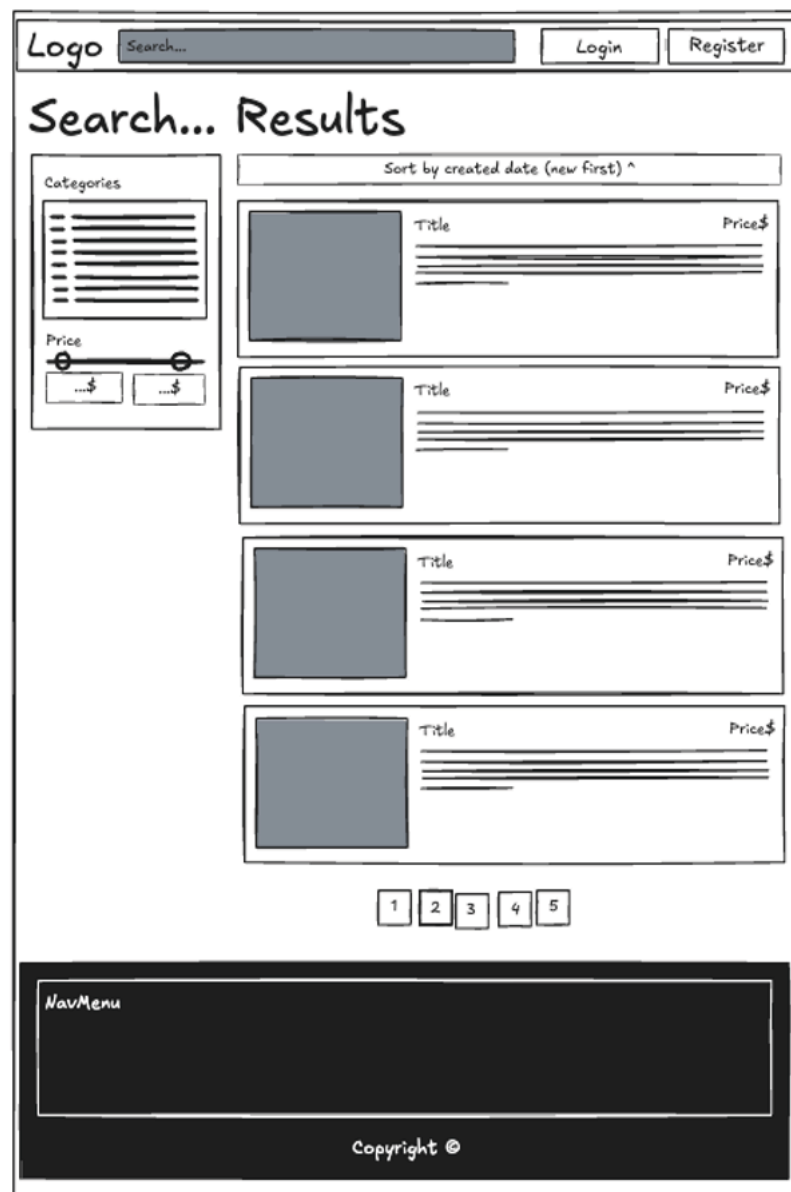


Рисунок 3.12 – Шаблон сторінки пошуку та фільтрації курсів

Сторінка покупки курсу (рис. 3.13) має такі елементи:

- хедер, поле для пошуку курсів та кнопки для реєстрації або авторизації;
- блок фільтрів;
- блок інформації курсу;
- блок додавання курсу до кошика або списку бажань;
- блок рекомендацій;
- футер, меню навігації.

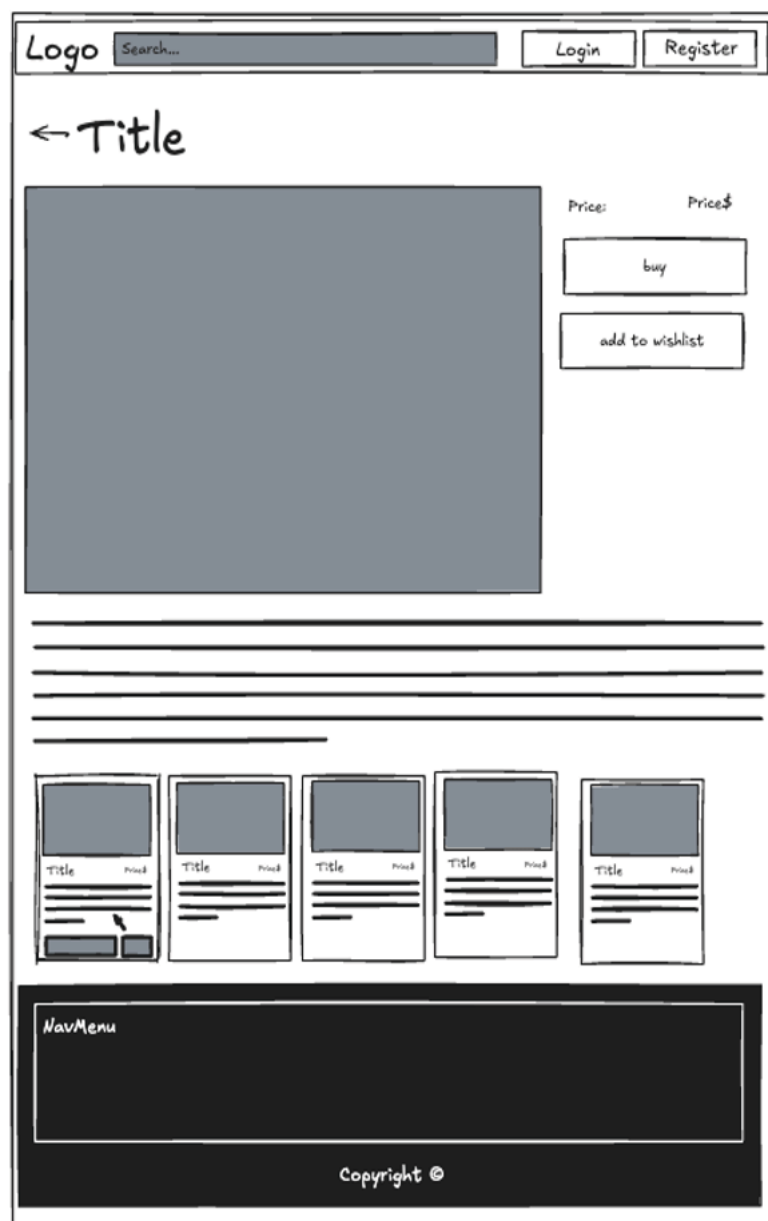


Рисунок 3.13 – Шаблон сторінки покупки курсу.

Сторінка списку бажань (рис. 3.14) має такі елементи:

- хедер, поле для пошуку курсів та кнопки для реєстрації або авторизації;
- блок списоку курсів у списку бажань;
- футер, меню навігації.

The wireframe illustrates the layout of a wishlist page. At the top, a header section contains a 'Logo' placeholder, a search bar with the text 'Search...', and two buttons labeled 'Login' and 'Register'. Below the header, the main content area is titled 'Wishlist'. It features a vertical list of six course items. Each item is represented by a horizontal box containing a placeholder image on the left, followed by a 'Title' label and three horizontal lines for text. To the right of the title is a 'Price\$' label, and further right is a button with a right-pointing arrow. A horizontal separator line is positioned between the fifth and sixth items. At the bottom of the page, a dark footer section contains a 'NavMenu' placeholder on the left and a 'Copyright ©' label on the right.

Рисунок 3.14 – Шаблон сторінки списку бажань

Сторінка корзини (рис. 3.15) має такі елементи:

- хедер, поле для пошуку курсів та кнопки для реєстрації або авторизації;
- блок списку курсів;
- кнопка для оформлення замовлення;
- футер, меню навігації.

The wireframe illustrates the layout of a shopping basket page. At the top, the header includes a 'Logo' placeholder, a 'Search...' input field, and 'Login' and 'Register' buttons. Below the header, the main content area is titled 'Basket'. It features a list of items, each represented by a row with a placeholder image, a 'Title' label, and a 'Price\$' label. To the right of the item list, a sidebar displays 'Total Price: Price\$' and a 'Checkout' button. The footer section contains a 'NavMenu' placeholder and a 'Copyright ©' notice.

Рисунок 3.15 – Шаблон сторінки корзини

Сторінка оформлення замовлення (рис. 3.16) має такі елементи:

- хедер, поле для пошуку курсів та кнопки для реєстрації або авторизації;
- блок форми оформлення замовлення;
- блок інформації щодо замовлення;
- кнопка оплати замовлення;
- футер, меню навігації.

The image shows a hand-drawn wireframe of a checkout page. At the top is a header bar containing a 'Logo' on the left, a search bar with the placeholder text 'Search...' in the center, and two buttons labeled 'Login' and 'Register' on the right. Below the header, the main content area is titled 'Checkout' in a large, bold font. This area is divided into two main sections. The left section is a form for user information and payment details, containing the following elements: a text input field for 'Email or phone number', a text input field for 'First name', a text input field for 'Last name', a 'Payment system' section with two buttons labeled 'Visa' (blue) and 'Mastercard' (yellow), a text input field for 'Card number', and two text input fields for 'Expired date' and 'CVV'. The right section is a summary box containing the text 'Total Price: Price\$' and a large grey button labeled 'Pay'. At the bottom of the page is a dark footer bar. On the left side of the footer is a box labeled 'NavMenu', and on the right side is the text 'Copyright ©'.

Рисунок 3.16 – Шаблон сторінки оформлення замовлення

Сторінка курсів користувача (рис. 3.17) має такі елементи:

- хедер, поле для пошуку курсів та кнопки для реєстрації або авторизації;
- список курсів користувача;
- футер, меню навігації.

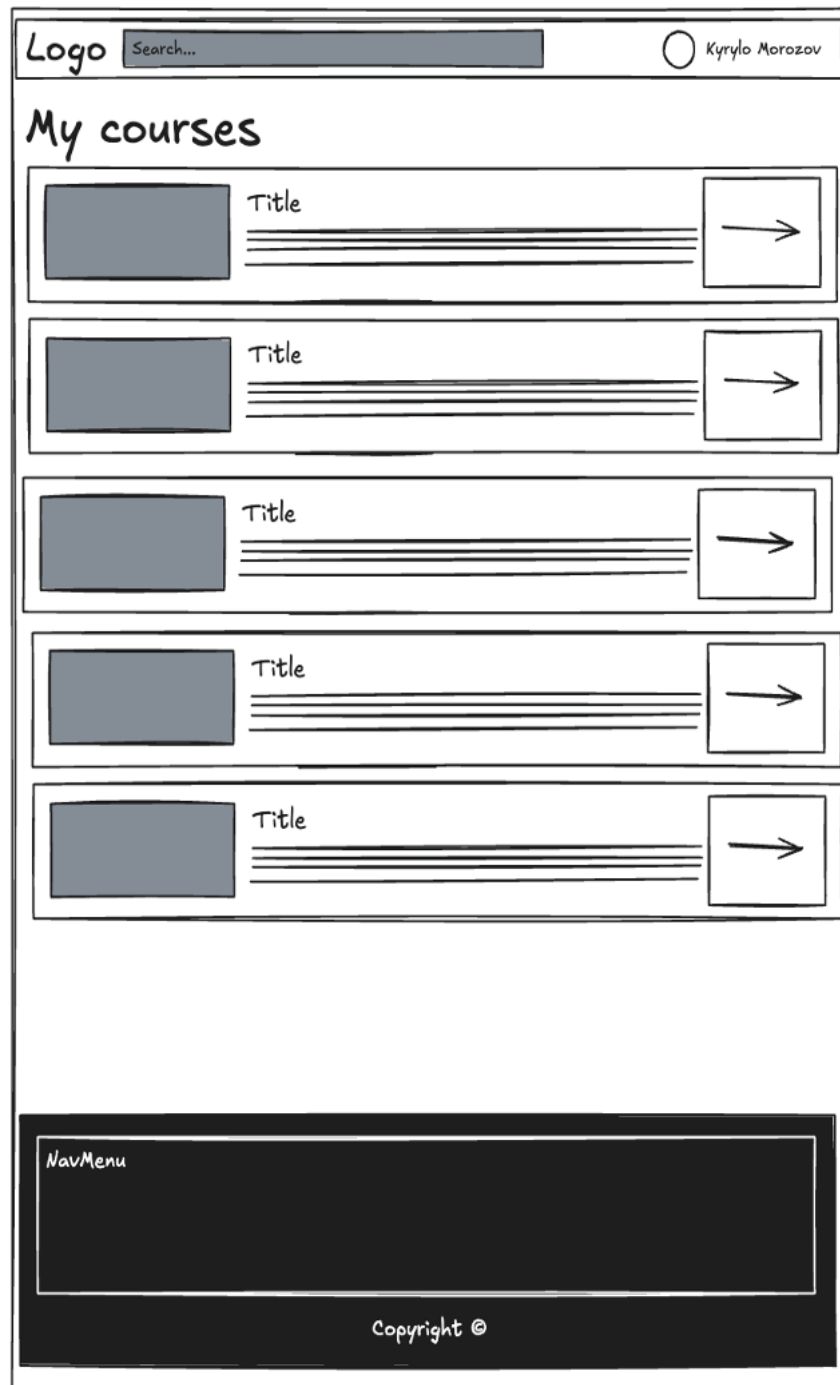


Рисунок 3.17 – Шаблон сторінки курсів користувача

Сторінка занять курсу (рис. 3.18) має такі елементи:

- хедер, поле для пошуку курсів та кнопки для реєстрації або авторизації;
- список занять курсу;
- футер, меню навігації.

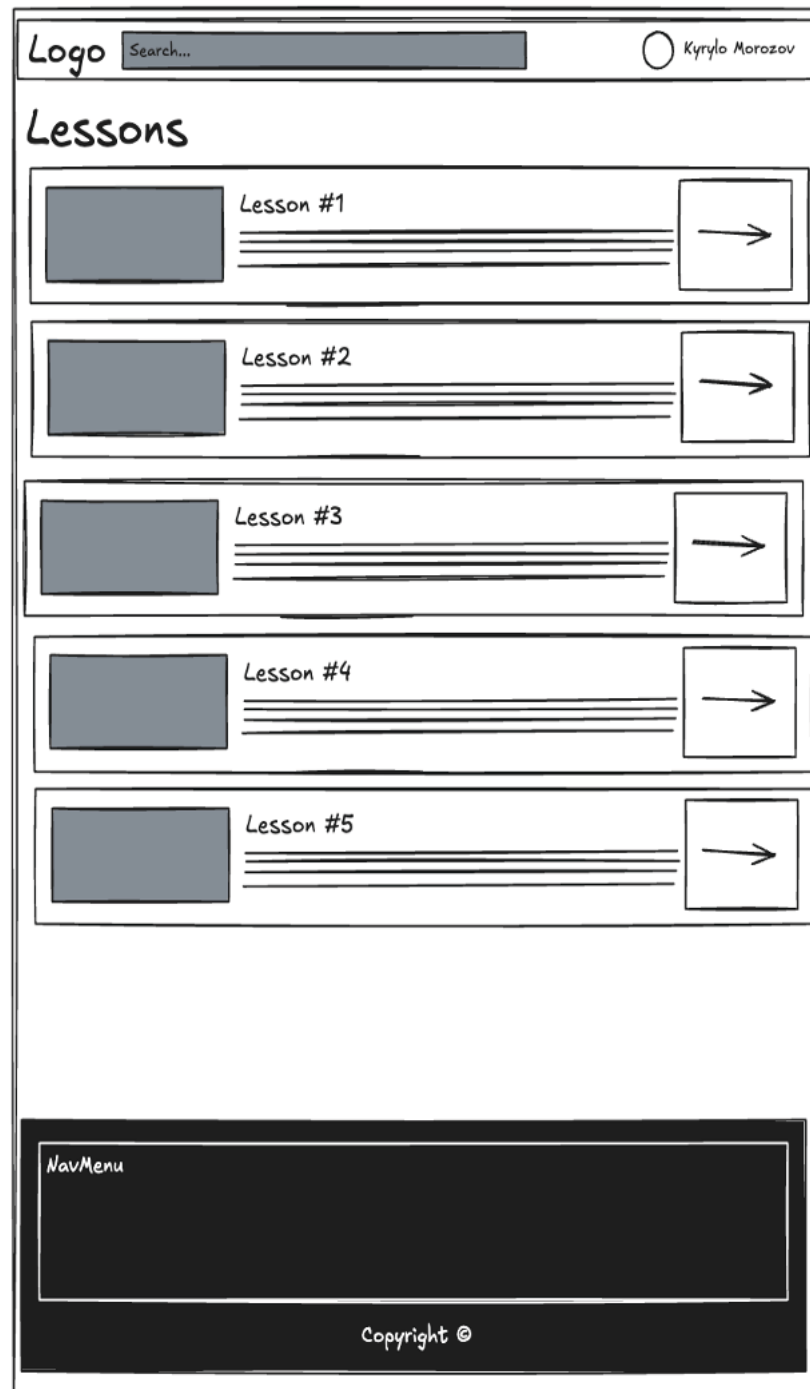


Рисунок 3.18 – Шаблон сторінки занять курсу

Сторінка заняття (рис. 3.19) має такі елементи:

- хедер, поле для пошуку курсів та кнопки для реєстрації або авторизації;
- кнопка повернення до списку занять;
- блок відео заняття;
- блок опису заняття;
- кнопка повернення до списку занять;
- кнопка відмітити курс як завершений;
- футер, меню навігації.

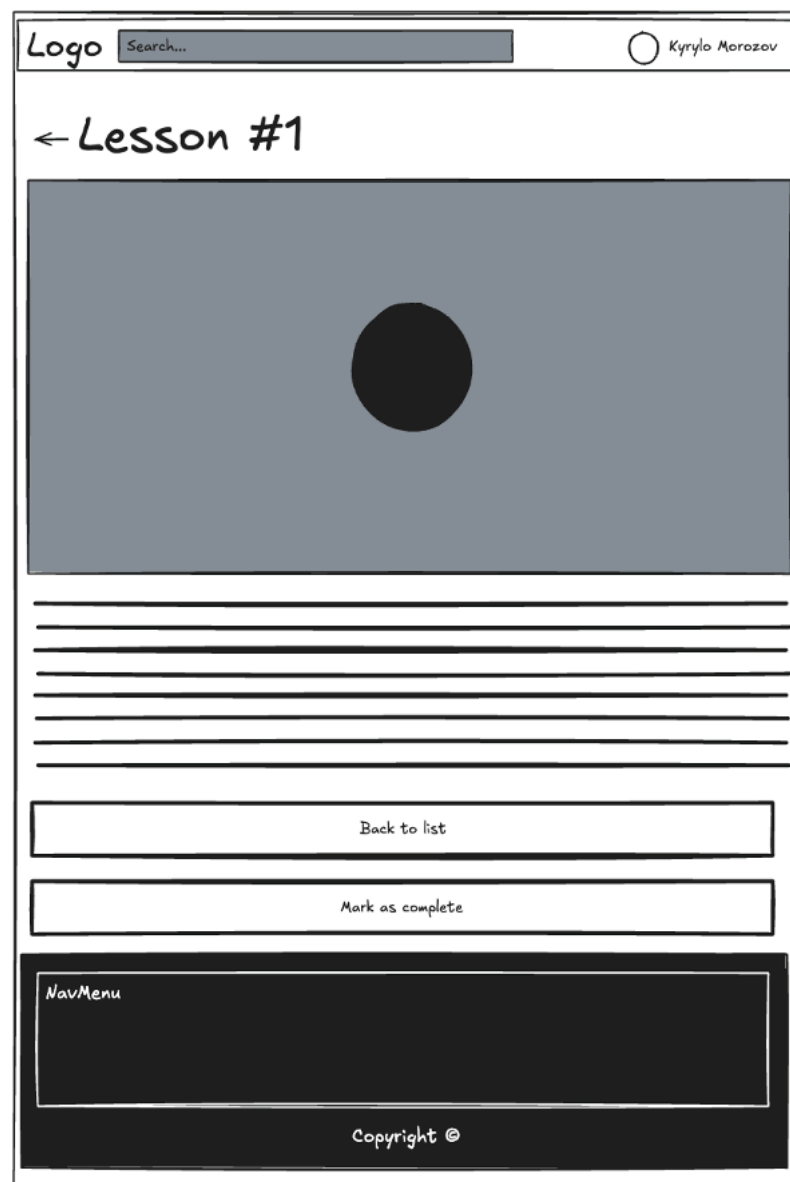


Рисунок 3.19 – Шаблон сторінки заняття

Сторінка списку курсів викладача (рис. 3.20) має такі елементи:

- хедер, поле для пошуку курсів та кнопки для реєстрації або авторизації;
- кнопка створення курсу;
- список курсів;
- кнопка видалення курсу;
- футер, меню навігації.

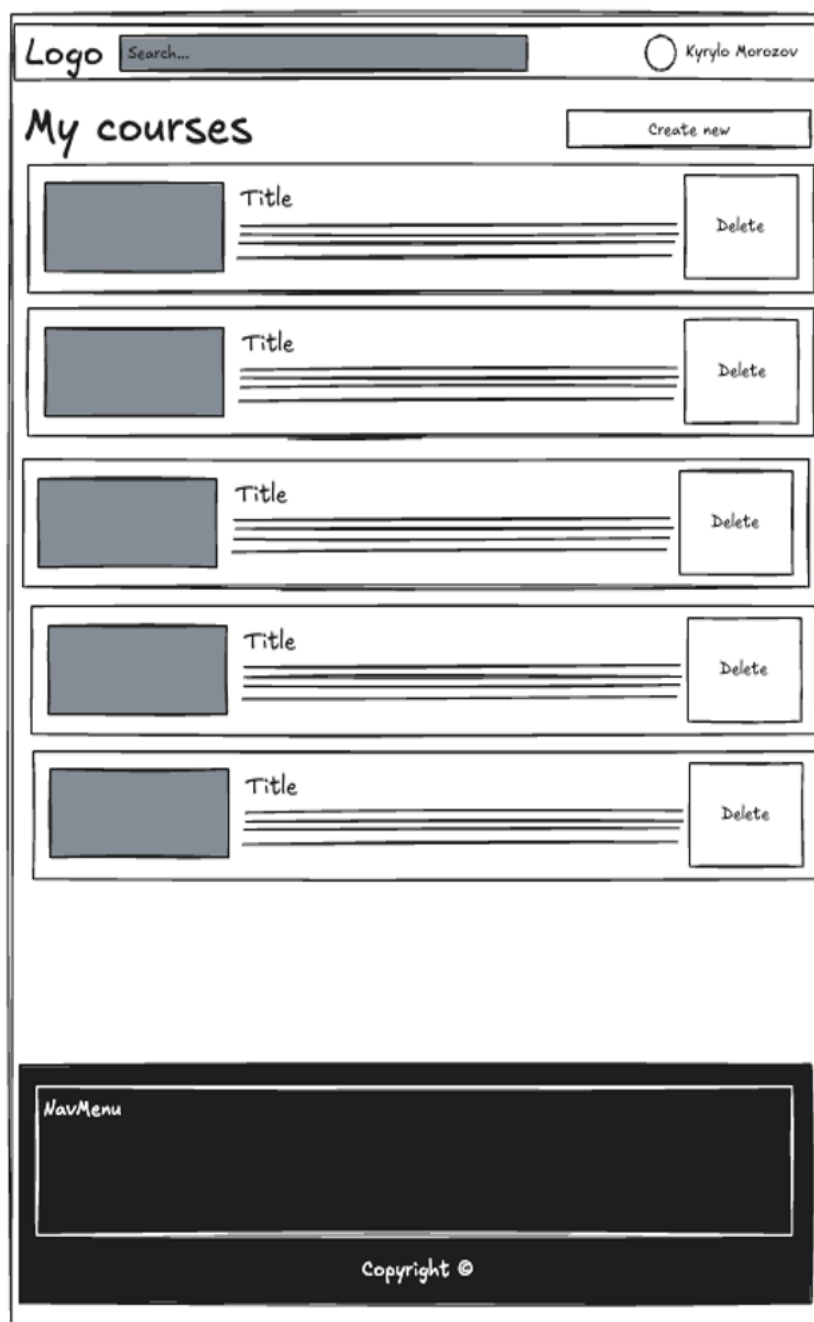


Рисунок 3.20 – Шаблон сторінки списку курсів викладача

Сторінка редагування або створення курсу (рис. 3.21) має такі елементи:

- хедер, поле для пошуку курсів та кнопки для реєстрації або авторизації;
- поле для редагування назви курсу;
- поле завантаження відео матеріалів курсу;
- поля опису курсу;
- кнопка створення нового заняття;
- список занять;
- футер, меню навігації.

The wireframe illustrates the layout of a course editing or creation page. At the top is a header bar containing a 'Logo' label, a search input field with the placeholder 'Search...', and a user profile section with a circular icon and the name 'Kyrylo Morozov'. Below the header, there is a large text input field labeled 'Title'. This is followed by a wide, light-gray rectangular area with a dark gray circle in the center containing the word 'Upload', intended for video uploads. Underneath this is a section with four horizontal lines for text input. A 'Save' button is positioned below these lines. The main content area is titled 'Lessons' and includes a 'Create new' button. It contains two lesson entries: 'Lesson #1' and 'Lesson #2'. Each entry features a small gray square icon, a text input field for the lesson title, and a right-pointing arrow button. At the bottom of the page is a dark gray footer bar with a 'NavMenu' label and a 'Copyright ©' notice.

Рисунок 3.21 – Шаблон сторінки редагування курсу

Сторінка редагування або створення заняття (рис. 3.22) має такі елементи:

- хедер, поле для пошуку курсів та кнопки для реєстрації або авторизації;
- поле для редагування назви заняття;
- поле завантаження відео матеріалів заняття;
- поля опису заняття;
- кнопка збереження змін;
- футер, меню навігації.

The wireframe illustrates the layout of a lesson editing or creation page. At the top, a header bar contains a 'Logo' on the left, a search input field labeled 'Search...' in the center, and a user profile icon with the name 'Kyrylo Morozov' on the right. Below the header, the main content area is titled 'Lesson #1' in a box. This is followed by a large gray rectangular area representing a video upload, which contains a central black circle with the word 'Upload' inside. Underneath the video area is a form for the lesson description, consisting of several horizontal lines. Below the description form is a 'Save' button. At the bottom of the page, there is a dark footer area containing a 'NavMenu' box and a 'Copyright ©' notice.

Рисунок 3.22 – Шаблон сторінки створення або редагування заняття

3.5.2 Розробка користувацького інтерфейсу

Оскільки шрифти безпосередньо впливають на сприйняття контенту користувачем, це є одним із відповідальних етапів. Для проєкту було обрано шрифти з Google Fonts [28], зокрема шрифт «Roboto» [29], який забезпечує високу читабельність, сучасний вигляд та універсальність. Цей шрифт легко адаптується до різних екранів і розмірів, що робить його ідеальним вибором для веб-дизайну.

Для вибору кольорової палітри для дизайну був використаний онлайн сервіс Coolors [30]. Цей сервіс є великим каталогом різноманітних палітр.

Вибрана палітра складається з фіолетового, чорного та білого кольорів. Фіолетовий колір надає дизайну елегантність і сучасність, чорний — контрастність і стійкість, а білий — легкість і чистоту. Така комбінація кольорів забезпечує гармонійне поєднання і покращує сприйняття контенту користувачем.

В якості іконок для проєкту були вибрані іконки з Lucide Icons [31], які забезпечують високу якість і чіткість візуальних елементів.

Lucide Icons [31] – це набір сучасних, мінімалістичних і універсальних іконок, який ідеально підходить для сучасних веб-дизайнів. Вибір цих іконок обумовлений їхнім легким, але виразним стилем, що добре поєднується з кольоровою палітрою і шрифтами. Іконки Lucide Icons [31] оптимізовані для використання в інтерфейсах, де важлива чіткість і розбірливість при будь-якому розмірі елемента.

Використання цих іконок дозволяє створити стильний і сучасний вигляд, що не лише додає естетичної цінності, але й покращує загальне враження від користувацького досвіду.

Для генерації логотипу був використаний сервіс Design.com [32], який дозволяє створити логотип на основі опису. Це дозволяє отримати унікальний та професійно виглядаючий логотип, що відповідає вимогам

бренду і сприяє зміцненню його впізнаваності. Логотип платформи зображено на рисунку 3.23.



Рисунок 3.23 – Логотип платформи

Дизайн користувацького інтерфейсу був спроектований за допомогою сервісу Uizard.io [33]. Цей інструмент дозволяє швидко створювати прототипи інтерфейсу по текстовому опису та макетах проєкту, надаючи можливість редагувати макети в реальному часі. Дизайн інтерфейсу зображено на рисунках 3.24, 3.25, 3.26, 3.27.

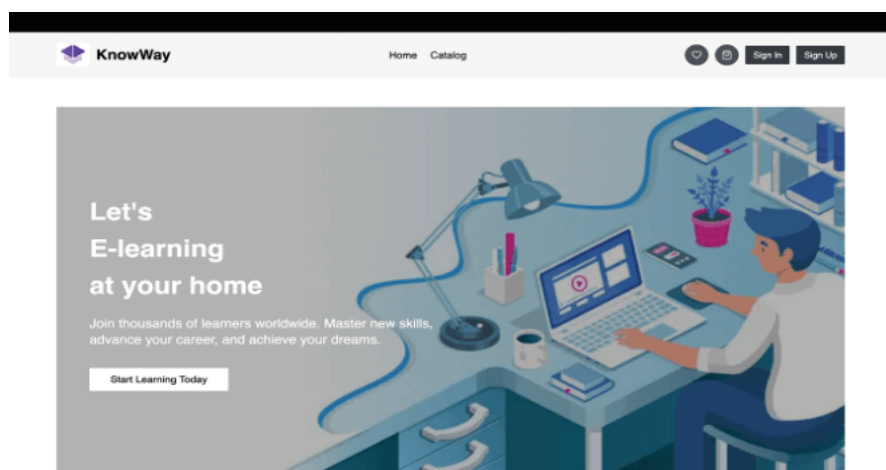


Рисунок 3.24 – Дизайн головної сторінки

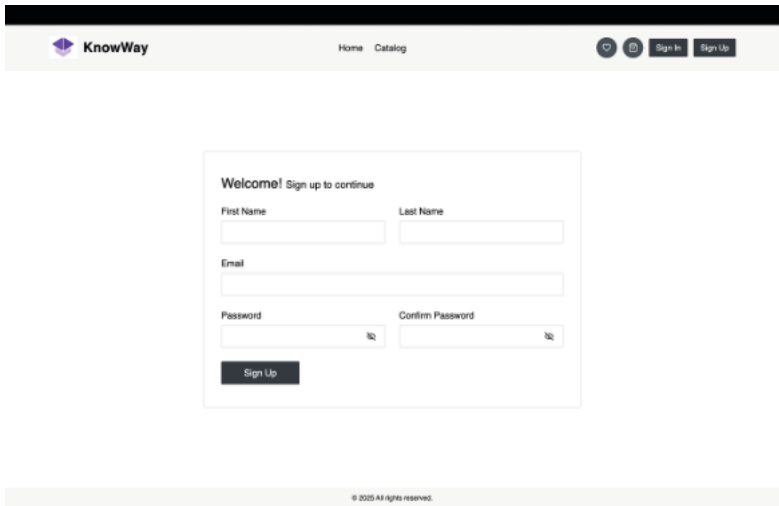


Рисунок 3.25 – Дизайн сторінки реєстрації

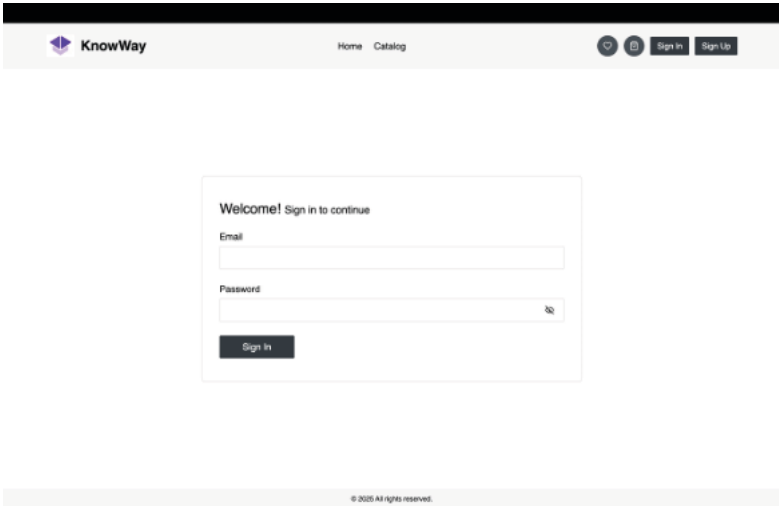


Рисунок 3.26 – Дизайн сторінки логіну

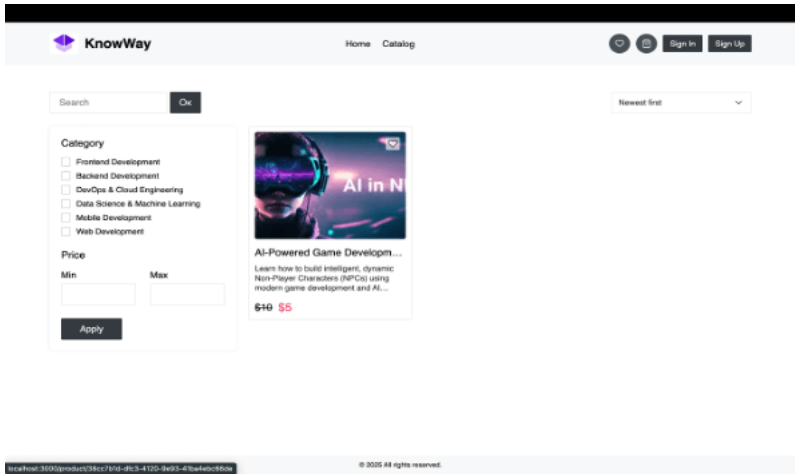


Рисунок 3.27 – Дизайн сторінки пошуку курсів

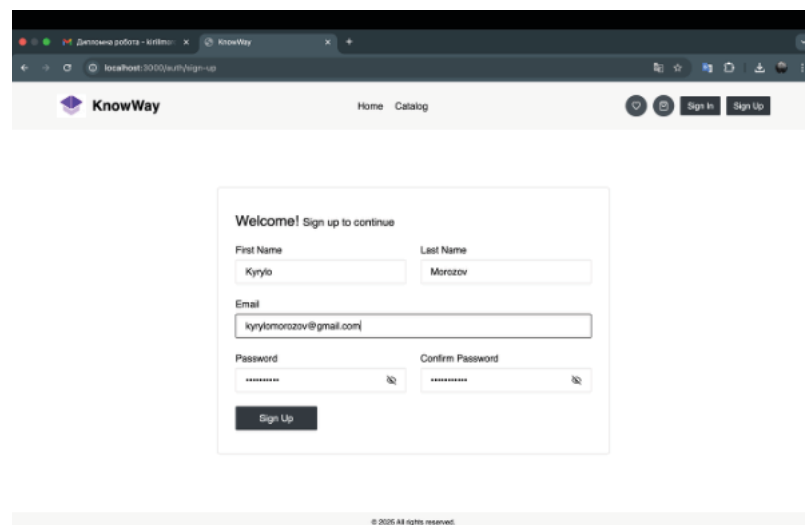
4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

Етапи експлуатації, тестування та експериментального дослідження програми є ключовими для забезпечення її надійності, продуктивності та відповідності вимогам користувачів.

Тестування дозволяє виявити та усунути можливі помилки, а експериментальне дослідження допомагає оцінити ефективність та поведінку системи в реальних умовах. Експлуатація передбачає практичне використання програми, що дозволяє отримати досвід використання і внести необхідні вдосконалення.

4.1 Опис роботи з програмою

Для запуску системи необхідно запустити вебзастосунок використовуючи команду `npm run start` і перейти за адресою `http://localhost:3000/`. Після завантаження користувачеві слід натиснути кнопку «Зареєструватися» та заповнити необхідні дані, такі як ім'я, електронна пошта та пароль. Процес реєстрації зображено на рисунку 4.1.



The screenshot displays a web browser window with the URL `localhost:3000/auth/sign-up`. The page header includes the 'KnowWay' logo, navigation links for 'Home' and 'Catalog', and user action buttons for 'Sign In' and 'Sign Up'. The main content area features a sign-up form titled 'Welcome! Sign up to continue'. The form contains input fields for 'First Name' (filled with 'Kyrylo'), 'Last Name' (filled with 'Morozov'), 'Email' (filled with 'kyrylomorozov@gmail.com'), 'Password', and 'Confirm Password'. A 'Sign Up' button is positioned at the bottom of the form. The footer of the page states '© 2025 All rights reserved.'

Рисунок 4.1 – Процес реєстрації нового користувача у системі

Якщо обліковий запис вже створений, можна увійти, використовуючи відповідні дані для авторизації натиснувши на кнопку «Увійти» (рис 4.2).

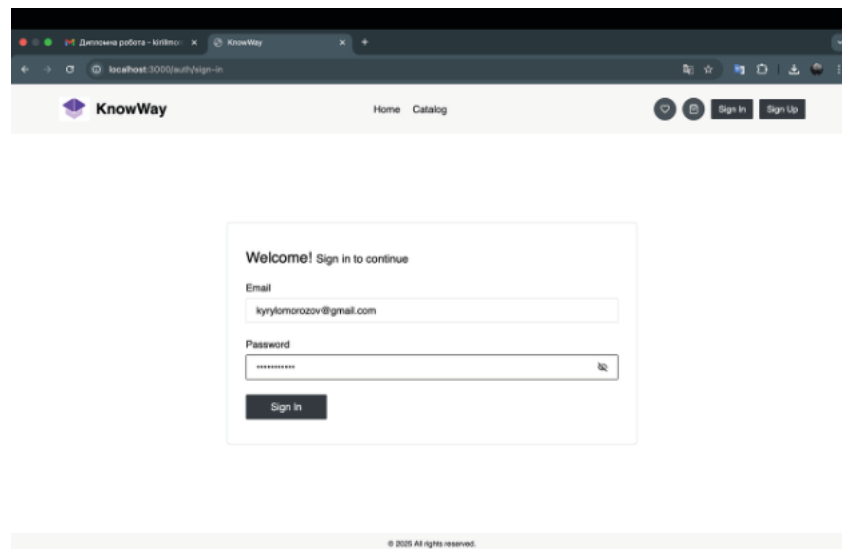


Рисунок 4.2 – Процес логіну користувача

Після успішної реєстрації або входу користувач автоматично потрапляє на головний екран платформи (рис. 4.3).

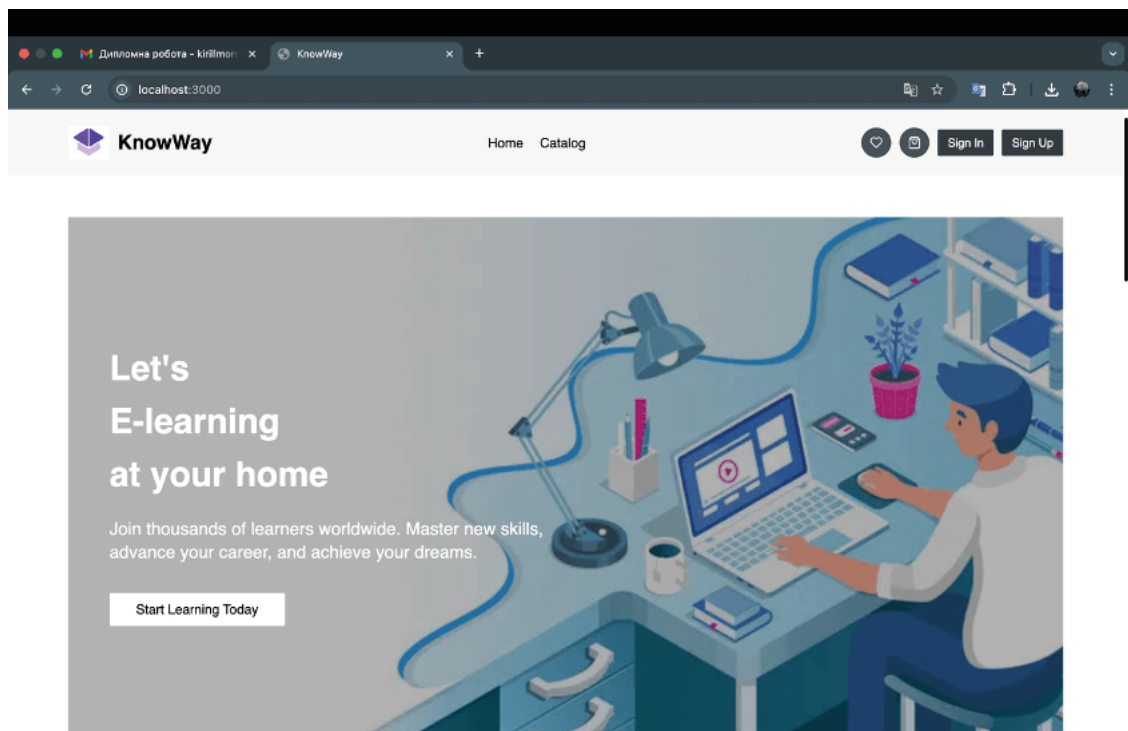


Рисунок 4.3 – Головна сторінка

На сторінці каталогу (рис. 4.4) користувач має можливість переглядати доступні курси або шукати конкретний курс через пошукове поле, розташоване у верхній частині сторінки. Для цього потрібно ввести ключові слова та натиснути кнопку “Пошук”. Система перенаправить користувача на сторінку з результатами, де буде представлено список курсів, що відповідають запиту.

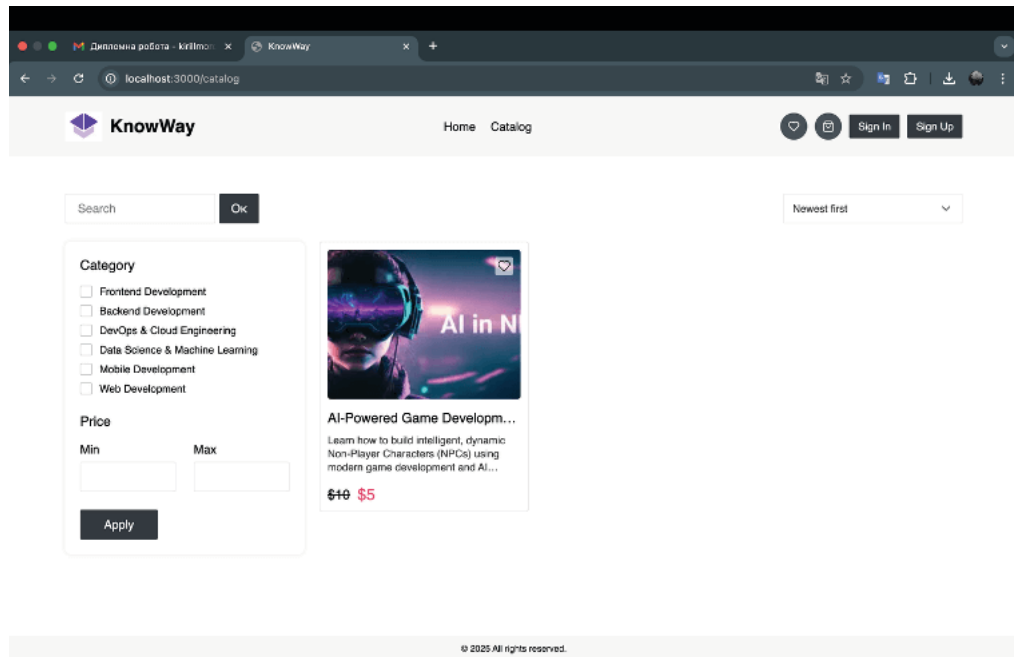


Рисунок 4.4 – Сторінка каталогу курсів

На сторінці результатів користувач може налаштувати відображення курсів, використовуючи доступні фільтри, які знаходяться в лівій частині екрана. Фільтрація можлива за категорією курсу або за вартістю. Крім того, під шапкою профілю розташовані елементи сортування, що дозволяють впорядкувати результати за датою створення або ціною.

Для придбання курсу необхідно обрати потрібний варіант із результатів пошуку та натиснути кнопку «Додати до кошика» (рис. 4.5). Після цього користувач переходить до сторінки оформлення замовлення, де йому потрібно ввести необхідні платіжні та контактні дані. Після успішної оплати курс додається до каталогу особистих курсів користувача.

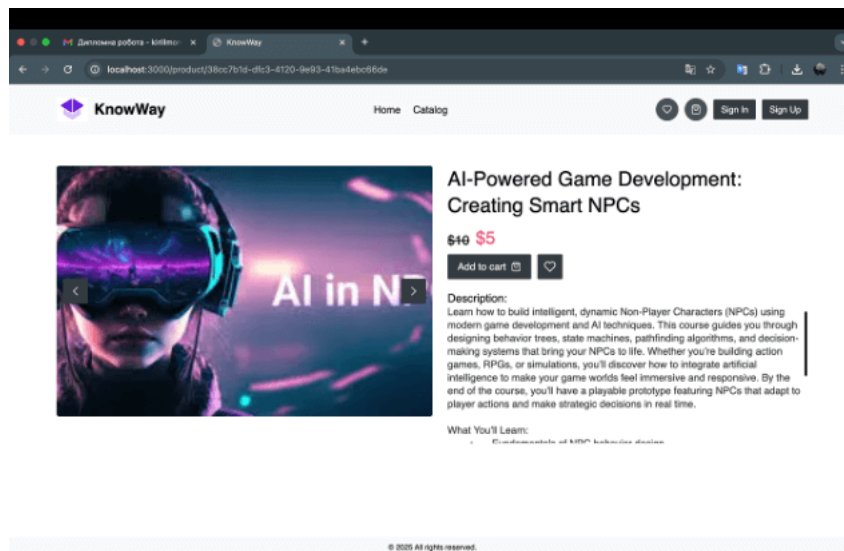


Рисунок 4.5 – Сторінка курсу

У каталозі особистих курсів користувач може переглядати придбані курси, проходити заняття та відзначати виконані модулі. Якщо потрібно почати курс заново, система надає можливість скинути поточний прогрес, натиснувши відповідний пункт у меню курсу. Також користувач може перемістити курс до архіву, якщо він тимчасово не планує його проходити. В архіві доступна функція розархівування, після чого курс знову з'явиться у каталозі активних курсів.

Крім основного функціоналу платформи, користувач має змогу редагувати свої особисті дані. Для цього потрібно натиснути на іконку користувача у верхній частині сторінки, після чого система перенаправить його до налаштувань профілю (рис. 4.6). Тут можна змінити пароль, оновити контактну інформацію та зберегти зміни, заповнивши відповідні форми.

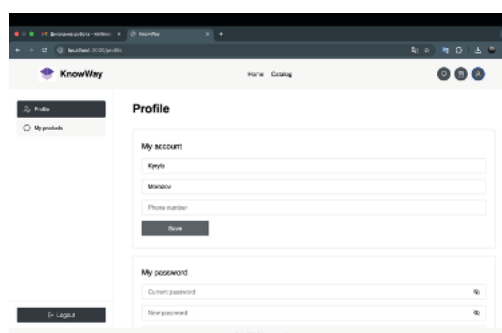


Рисунок 4.6 – Сторінка налаштувань особистих даних

4.2 Тестування програмного забезпечення

Для тестування програмного забезпечення було обрано ручний метод, за якого користувач тестувальник перевіряє весь наявний функціонал програмного забезпечення без використання автоматизованих інструментів, та заносить дані результатів тестування до спеціальної таблиці. Цей підхід було обрано через його простоту та зручність у реалізації. Алгоритм тестування наведено у таблиціях 4.1, 4.2, 4.3, 4.4.

Таблиця 4.1 – Тестування процесу реєстрації користувача з невалідними даними

Назва тесту	Реєстрація користувача з невалідними даними	
Номер кроку	Опис дій кроку	Результат
1	Відкрити сторінку реєстрації користувача	Відкрилась сторінка з формою реєстрації
2	Заповнити поле first name	-
3	Заповнити поле last name	-
4	Заповнити поле email	-
5	Заповнити поле password	-
6	Натиснути кнопку register	Користувачу вивело повідомлення про невалідні дані

Таблиця 4.2 – Тестування процесу реєстрації користувача з валідними даними

Назва тесту	Реєстрація користувача з валідними даними	
Номер кроку	Опис дій кроку	Результат
1	Відкрити сторінку реєстрації користувача	Відкрилась сторінка з формою реєстрації

Продовження таблиці 4.2

Назва тесту	Реєстрація користувача з валідними даними	
Номер кроку	Опис дій кроку	Результат
2	Заповнити поле first name	-
3	Заповнити поле last name	-
4	Заповнити поле email	-
5	Заповнити поле password	-
6	Натиснути кнопку register	Користувача перенаправило на головну сторінку та зареєструвало в системі

Таблиця 4.3 – Тестування процесу логіну користувача з невалідними даними

Назва тесту	Логін користувача з невалідними даними	
Номер кроку	Опис дій кроку	Результат
1	Відкрити сторінку логіну користувача	Відкрилась сторінка з формою логіну
2	Заповнити поле email	-
3	Заповнити поле password	-
4	Натиснути кнопку login	Користувачу вивело повідомлення про невалідні дані

Таблиця 4.4 – Тестування процесу логіну користувача з валідними даними

Назва тесту	Логін користувача з валідними даними	
Номер кроку	Опис дій кроку	Результат
1	Відкрити сторінку логіну користувача	Відкрилась сторінка з формою логіну

Продовження таблиці 4.4

Назва тесту	Логін користувача з валідними даними	
Номер кроку	Опис дій кроку	Результат
2	Заповнити поле email	-
3	Заповнити поле password	-
4	Натиснути кнопку login	Користувача перенаправило на головну сторінку

ВИСНОВКИ

У цій роботі досліджено розробку онлайн платформи для онлайн освіти. Розроблений вебзастосунок дозволяє користувачам проходити онлайн курси та отримувати сертифікати після їх проходження.

Цінність проєкту полягає в покращенні якості та доступності освіти для користувачів з різних регіонів.

Дослідження включало аналіз існуючих освітніх платформ, виявлення їхніх переваг і недоліків та головних факторів успішності та популярності існуючих платформ. Що дозволило визначити основні функціональні і нефункціональні вимоги, а також визначити головну аудиторію вебзастосунку.

Результатами дослідження є вебзастосунок, який забезпечує можливість онлайн навчання завдяки курсам для користувачів. Завдяки чому він може застосовуватись у різних освітніх сферах: корпоративному навчанні, професійній підготовці або саморозвитку.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Udemy [Electronic resource]. – Access mode: <https://www.udemy.com/>
2. Coursera [Electronic resource]. – Access mode: <https://www.coursera.org/>
3. Educative [Electronic resource]. – Access mode: <https://www.educative.io/>
4. How to Choose the Best Colors For Logo Design – Emotion of Colors [Electronic resource]. – Access mode: <https://www.logodesignteam.com/blog/best-colors-for-logo-design/>
5. TypeScript [Electronic resource]. – Access mode: <https://www.TypeScriptlang.org/>
6. JavaScript [Electronic resource]. – Access mode: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
7. Angular [Electronic resource]. – Access mode: <https://angular.dev/>
8. NestJs [Electronic resource]. – Access mode: <https://nestjs.com/>
9. NextJs [Electronic resource]. – Access mode: <https://nextjs.org/>
10. The 2020 Stack Overflow Developer Survey – 65,000 Devs Share Their Salaries, Top Programming Languages, and More [Electronic resource]. – Access mode: <https://www.freecodecamp.org/news/stack-overflow-developer-survey-2020-programming-language-framework-salary-data/>
11. The 2020 Stack Overflow Developer Survey – 65,000 Devs Share Their Salaries, Top Programming Languages, and More [Electronic resource]. – Access mode: <https://www.freecodecamp.org/news/stack-overflow-developer-survey-2020-programming-language-framework-salary-data/>
12. Visual Studio Code [Electronic resource]. – Access mode: <https://code.visualstudio.com/>
13. Git [Electronic resource]. – Access mode: <https://git-scm.com/>

14. IntelliSense [Electronic resource]. – Access mode:
<https://code.visualstudio.com/docs/editor/intellisense>
15. WebStorm [Electronic resource]. – Access mode:
<https://www.jetbrains.com/webstorm/>
16. Atom [Electronic resource]. – Access mode: <https://atom-editor.cc/>
17. Sublime Text [Electronic resource]. – Access mode:
<https://www.sublimetext.com/>
18. Eclipse [Electronic resource]. – Access mode:
<https://www.eclipse.org/>
19. Java [Electronic resource]. – Access mode: <https://www.java.com/>
20. Top 4 TypeScript Frameworks [Electronic resource]. – Access
mode: <https://verpex.com/blog/website-tips/top-4-TypeScript-frameworks>
21. NuxtJs [Electronic resource]. – Access mode: <https://nuxt.com/>
22. jQuery [Electronic resource]. – Access mode: <https://jquery.com/>
23. Svelte [Electronic resource]. – Access mode: <https://svelte.dev/>
24. React [Electronic resource]. – Access mode: <https://react.dev/>
25. ExpressJs [Electronic resource]. – Access mode:
<https://expressjs.com/>
26. PostgreSQL [Electronic resource]. – Access mode:
<https://www.postgresql.org/>
27. TypeORM [Electronic resource]. – Access mode:
<https://typeorm.io/>
28. Google Fonts [Electronic resource]. – Access mode:
<https://fonts.google.com/>
29. Font Roboto [Electronic resource]. – Access mode:
<https://fonts.google.com/specimen/Roboto>
30. Coolors [Electronic resource]. – Access mode: <https://coolors.co/>
31. Lucide Icons [Electronic resource]. – Access mode:
<https://lucide.dev/icons/>

32. Design.com [Electronic resource]. – Access mode: <https://www.design.com/>
33. Uizard [Electronic resource]. – Access mode: <https://uizard.io/>
34. Top 4 TypeScript Frameworks [Electronic resource]. – Access mode: <https://verpex.com/blog/website-tips/top-4-TypeScript-frameworks>

ДОДАТОК А
Технічне завдання

A.1 Вступ

Це технічне завдання присвячене розробці програмного забезпечення, яке має на меті оптимізувати та цифровізувати процес сертифікації знань, а також створити зручну платформу для дистанційного онлайн-навчання.

Застосунок повинен забезпечити інтерактивну взаємодію між студентами та викладачами, полегшити доступ до освітніх матеріалів і надати сучасні інструменти для самостійного навчання та професійного зростання.

A.2 Підстави для розробки

Основою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Вебзастосунок для дистанційного онлайн-навчання», затверджене наказом Національного університету «Запорізька політехніка» № 209 від 28 квітня 2025 року.

A.3 Призначення розробки

Головна мета програмного продукту — розробка ефективного цифрового засобу для онлайн-освіти та сертифікації.

Застосунок надає користувачам можливість шукати, проходити та купувати онлайн-курси, що допомагає підвищувати професійні навички, необхідні як для початку кар'єри, так і для її подальшого розвитку.

Для викладачів система виступає платформою для створення й просування власного навчального контенту, а також для аналізу ефективності освітнього процесу.

А.3 Основні вимоги до програми

А.3.1 Вимоги до функціональних характеристик

Вимоги до функціональних характеристик складаються з наступних:

Для студентів:

- перегляд списку необхідних навичок для популярних професій;
- додавання курсів до списку бажань;
- фільтрація курсів за: мовою, ціною, темами, категоріями;
- пошук курсів за назвою;
- сортування курсів за: датою додавання (нові-старі, старі-нові), ціною (дешеві-дорогі, дорогі-дешеві);
- збереження прогресу проходження курсів;
- завантаження додаткових матеріалів до уроків;
- скидання прогресу проходження курсів;
- переміщення курсів до архіву;
- відновлення курсів з архіву;
- автоматичне архівування курсів після завершення;
- можливість купувати курси для відкриття доступу до них.

Для викладачів:

- створення власних курсів;
- перегляд аналітики та метрик курсів;
- перегляд аналітики продажів курсів;
- приховування курсів з пошуку;
- видалення курсів;
- редагування курсів.

А.3.2 Вимоги до надійності

Надійна робота системи повинна забезпечуватися за рахунок:

- стабільної роботи серверної інфраструктури;

- використання перевіреної системи керування базами даних;
- коректного оброблення вхідних даних користувача з урахуванням типізації для запобігання системним помилкам;
- реалізації механізмів резервного копіювання та захисту даних.

A.3.3 Умови експлуатації

Для надійної роботи вебзастосунку слід використовувати серверну інфраструктуру, яка відповідає очікуваному навантаженню – це може бути як хмарне середовище, так і виділений сервер. Система повинна забезпечувати високу швидкість обробки користувацьких запитів, гарантувати конфіденційність персональних даних та цілодобову доступність сервісу.

A.3.4 Вимоги до складу та параметрів технічних засобів

Вимоги до технічних засобів:

- процесор Intel Core i3-8100F з тактовою частотою, ГГц 3,6 або еквівалентний;
- оперативна пам'ять: DDR3 з об'ємом мінімум 4 Гб;
- засоби виводу інформації: монітор з роздільною здатністю не менше 1280x800 пікселів;
- дисковий простір: мінімум 1 Гб вільного місця на жорсткому диску.

Вимоги до програмного забезпечення:

- веббраузер: останні версії Google Chrome, Mozilla Firefox, Microsoft Edge, Safari або аналогічних;
- інтернет-з'єднання: Стабільне з'єднання з Інтернетом для завантаження та використання вебзастосунку;
- підтримка JavaScript: увімкнена підтримка веббраузером для виконання скриптів JavaScript;

– операційна система: Windows 10 або вище, macOS 10.13 або вище, Linux (Ubuntu, Fedora тощо).

A.4 Порядок контролю та приймання

Науковий керівник здійснює нагляд за виконанням дипломної кваліфікаційної роботи, контролюючи відповідність проміжних етапів поставленому технічному завданню. Остаточну оцінку проєкту та рішення щодо його прийняття ухвалює екзаменаційна комісія під час публічного захисту дипломної роботи.

ДОДАТОК Б
Текст програми

Б.1 Текст програми

Вихідний код цього програмного забезпечення є відкритим і доступний на платформі GitHub у формі публічних репозиторіїв. Для зручності структуру проєкту поділено на дві основні частини: серверну та клієнтську.

Серверна частина вебзастосунку містить серверну логіку, API та взаємодію з базою даних. Ознайомитися з кодом можна за наступним посиланням: <https://github.com/ReZerVV/lms-be>.

Клієнтська частина вебзастосунку відповідає за інтерфейс користувача та його взаємодію з системою. Весь відповідний код доступний за посиланням: <https://github.com/ReZerVV/lms-fe>.

ДОДАТОК В
Слайди презентації

В.1 Слайди презентації



Рисунок В.1 – Слайд №1

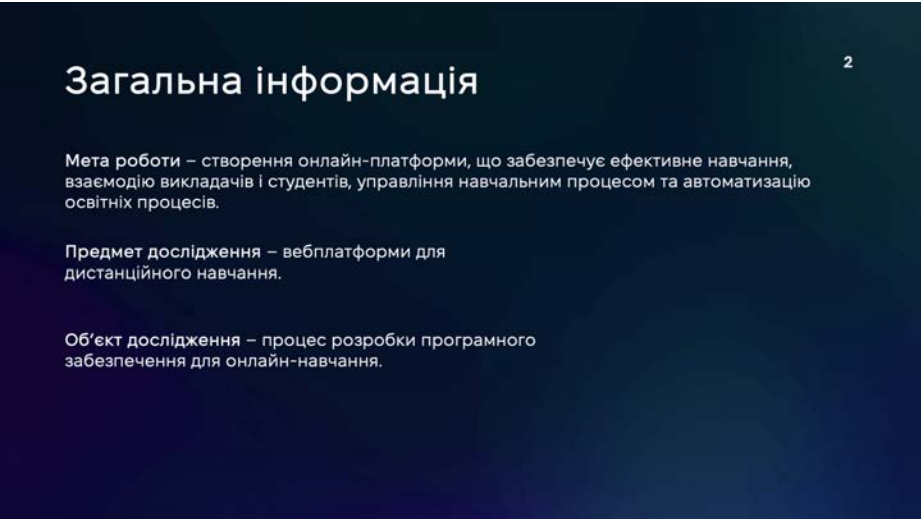


Рисунок В.2 – Слайд №2



Рисунок В.3 – Слайд №3

Планування застосунку

Вибір фреймворку програмування

4

Фреймворк	Переваги	Недоліки
Next.js	Серверний рендеринг (SSR), статична генерація (SSG), чудова SEO-оптимізація, продуктивність, проста інтеграція	Вимагає певного рівня знань React [24].
Nuxt.js	Аналог Next.js для Vue.js, підтримує SSR та SSG, зручний для Vue-розробників	Менша популярність у порівнянні з React, складніші налаштування
Angular	Повноцінний фреймворк з великою екосистемою підтримки TypeScript, двостороннє зв'язування даних	Високий поріг входу складна структура
jQuery	Простота у використанні велика кількість плагінів	Застарілий підхід, низька продуктивність для сучасних SPA-додатків
Svelte	Висока продуктивність, компілюється у звичайний JavaScript, простий у використанні	Менша екосистема та підтримка у порівнянні з React або Vue

Проаналізувавши найбільш популярні фреймворки для створення інтерфейсу, було розглянуто Next.js, Nuxt.js, Angular, jQuery та Svelte. Таблиця порівняння демонструє, що кожен з них має свої переваги та недоліки. З огляду на ефективність, SEO-оптимізацію, підтримку SSR та глибоку інтеграцію з TypeScript, для реалізації клієнтської частини проекту було обрано Next.js.

Рисунок В.4 – Слайд №4

Планування застосунку

Діаграма прецедентів

5

```
graph TD
    User((User)) --> Login[Login]
    User --> Register[Register]
    User --> Logout[Logout]
    Admin((Admin)) --> AddItem[Add Item]
    Admin --> EditItem[Edit Item]
    Admin --> DeleteItem[Delete Item]
    System((System)) --> ViewItem[View Item]
    System --> SearchItem[Search Item]
    User --> AddItem
    User --> EditItem
    User --> DeleteItem
    User --> ViewItem
    User --> SearchItem
```

Діаграма прецедентів — це візуальне представлення взаємодії користувачів (акторів) із системою через різні сценарії використання (прецеденти). Вона допомагає зрозуміти, які функції повинна виконувати система з точки зору кінцевого користувача.

Рисунок В.5 – Слайд №5

Планування застосунку

База даних

6

```
graph LR
    User[User] --> Item[Item]
    Item --> Category[Category]
    Item --> Location[Location]
```

ER-діаграма (діаграма сутність-зв'язок) — це схема, яка відображає структуру бази даних шляхом моделювання сутностей (таблиць), їхніх атрибутів (полів) та зв'язків між ними. Вона дозволяє візуально представити, які об'єкти існують у системі, як вони пов'язані між собою та які дані зберігаються.

Рисунок В.6 – Слайд №6

Планування застосунку

Архітектура додатку

Архітектура додатку — це структура, яка визначає, як компоненти системи взаємодіють між собою, яким чином організовані дані, логіка, інтерфейс та зовнішні сервіси. Вона описує принципи розподілу обов'язків між модулями, вибір шаблонів проектування, технологій та способів комунікації між частинами системи. Правильно побудована архітектура є критично важливою для масштабованості, підтримки, безпеки та продуктивності застосунку.

Рисунок В.7 – Слайд №7

Дизайн застосунку

Користувачський потік

Потік користувача (user flow) — це візуальне або текстове представлення кроків, які проходить користувач для досягнення певної цілі в додатку чи на вебсайті. Він демонструє логіку навігації між екранами, враховуючи можливі дії та сценарії використання. Добре продуманий потік користувача сприяє підвищенню зручності, зменшенню кількості помилок та загальному задоволенню від використання додатку.

Рисунок В.8 – Слайд №8

Дизайн застосунку

Шаблон дизайну

Макети інтерфейсу користувача — це візуальні схеми або прототипи, які демонструють зовнішній вигляд і розташування елементів на екранах додатку. Вони допомагають зрозуміти, як користувач буде взаємодіяти з продуктом, ще до початку розробки. Добре продумані макети сприяють ефективній командній роботі та економлять час і ресурси на етапі реалізації.

Рисунок В.9 – Слайд №9

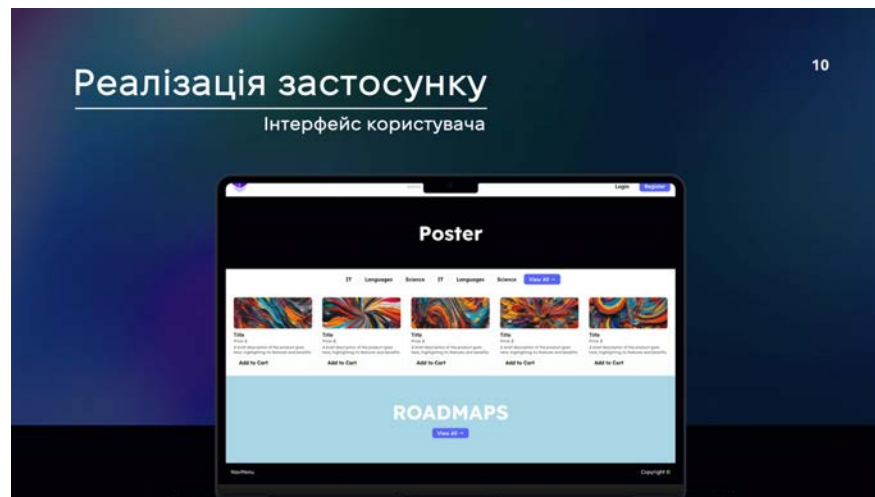


Рисунок В.10 – Слайд №10

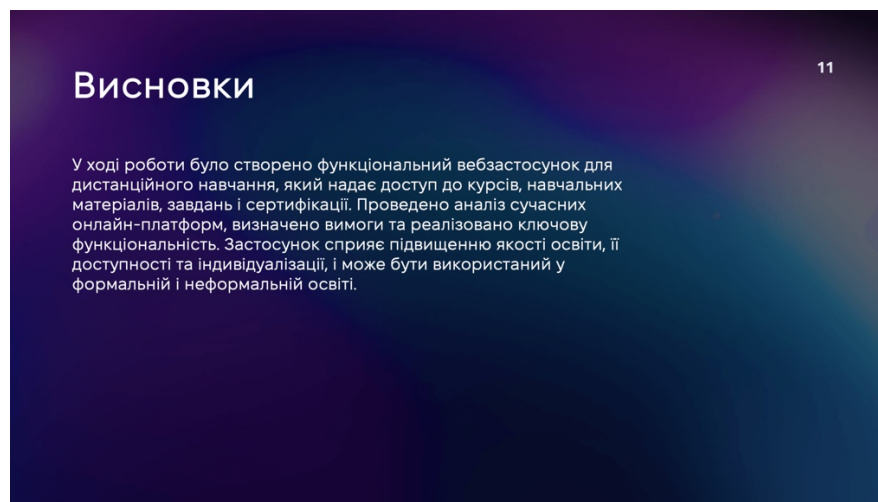


Рисунок В.11 – Слайд №11