

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»**

Спеціальність: 123 «Комп'ютерна інженерія»

Освітня програма: «Комп'ютерна графіка і Web-дизайн»

Група: 4КГ-06

Дипломний проект

**здобувача освіти денної форми навчання
КГ.06.21.000.ДП**

***МУРЗАК ІРИНИ
ВОЛОДИМИРІВНИ***

**м. Одеса
2023 р.**

Спеціальність: 123 «Комп'ютерна інженерія»

Освітня програма: «Комп'ютерна графіка і Web-дизайн»

Група: 4КГ-06

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи) на тему:

Розробка та впровадження адаптивного веб-сайту онлайн-навчання

Проектний матеріал складається з пояснювальної записки на 72 сторінках та графічного (презентаційного) матеріалу на 14 аркушах (слайдах).

Дипломник  (Мурзак І.В.)

Керівник  (Кунуп Т.В.)

Консультанти:

з економічної частини  (Копайгородська Т.Г.)

з охорони праці  (Чорновол Н.І.)

з дотримання вимог ЄСКД  (Петрашова В.І.)

старший консультант  (Кривченко А.А.)

До захисту допущений

Голова циклової комісії  (Кривченко Ю.В.)

Завідувач відділення  (Скорнякова О.В.)

Захист «23» сервіс 2023 р.

Протокол ДКК № 5

Оцінка ДКК 5 (відмінно)

Секретар ДКК 

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ і ПІ
Спеціальність 123 «Комп'ютерна інженерія»
Освітня програма «Комп'ютерна графіка і Web-дизайн»

ЗАТВЕРДЖУЮ:

Заст. директора з НВР


Беркань І.В.

“ ” 2023 р.

ЗАВДАННЯ

на дипломний проект (роботу)

Здобувчів(здобувачці освіти) Мурзак Ірині Володимирівні
(прізвище, ім'я, по батькові)

- Тема проекту (роботи) Розробка та впровадження адаптивного веб-сайту онлайн-навчання
затверджена наказом по коледжу від “17” лютого 2023 р. № 235-А2-ОД
- Термін здачі студентом закінченого проекту (роботи) 12.06.23
- Вихідні дані до проекту (роботи) Обґрунтування технології програмування. Розробка структури веб-сайту. Розробка дизайну веб-сайту. Мови розробки веб-сайту, використані фреймворки: HTML, Tailwind CSS, JavaScript, Vue.js, Laravel. Середовище розповсюдження: Internet.
- Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
Аналіз існуючих програмних рішень; Обґрунтування вибору програмного забезпечення для реалізації проекту; Формування вимог до веб-орієнтованої системи; Порядок розробки веб-сайту; Проектування структури веб-сайту; Розробка модульної сітки; Розробка прототипів; Розробка графічного дизайну; Проектування моделі бази даних; Інсталяція програмних залежностей та розробка клієнтської частини веб-сайту; Розробка серверної частини веб-сайту; Тестування розробленого сайту; Економічна частина; Охорона праці.
- Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Актуальність та мета дослідження; Основні завдання дипломної роботи; Аналіз існуючих програмних рішень; Структура веб-сайту; Вибір програмного забезпечення; Архітектура MVC; Формування вимог до веб сайту. User Story; Методологія атомарного дизайну; ER-діаграма; Результати тестування веб-сайту.

Консультанти по проекту (роботі), із зазначенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Розділи ДП	Кунуп Т.В.		
Економічні розрахунки	Копайгородська Т.Г.		
Охорона праці	Чорновол Н.І.		
ЄСКД	Петрашова В.І.		

7. Дата видачі завдання 01.05.2023

Керівник

Кунуп Т.В.

(підпис)

Завдання прийняв до виконання

(підпис)

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вступ. Постановка мети та задач проектування	22.05.2023 р.	виконана
2	Аналіз існуючих програмних рішень	23.05.2023 р.	виконана
3	Обґрунтування вибору програмного забезпечення для реалізації проекту	24.05.2023 р.	виконана
4	Порядок розробки веб-сайту	25.05.2023 р.	виконана
5	Розробка модульної сітки	26.05.2023 р.	виконана
6	Розробка прототипу	29.05.2023 р.	виконана
7	Розробка графічного дизайну	30.05.2023 р.	виконана
8	Інсталяція програмних залежностей та розробка клієнтської частини веб-сайту	01.06.2023 р.	виконана
9	Розробка серверної частини веб-сайту	03.06.2023 р.	виконана
10	Тестування розробленого сайту	05.06.2023 р.	виконана
11	Економічні розрахунки та питання з охорони праці	07.06.2023 р.	виконана
12	Аналіз результатів, створення слайдів презентації	09.06.2023 р.	виконана
13	Підготовка проекту до захисту	11.06.2023 р.	виконана

Дипломник

(підпис)

Керівник проекту

(підпис)

ЗМІСТ

ВСТУП	7
1 ТЕХНОЛОГІЧНИЙ РОЗДІЛ.....	9
1.1 Аналіз існуючих програмних рішень.....	9
1.2 Вибір програмного забезпечення для реалізації проекту інформаційної системи.....	14
1.2.1 Середовище розробки Visual Studio Code	14
1.2.2 Інструмент для створення інтерфейсу веб сайту Figma.....	16
1.2.3 Фреймворк Vue.js для клієнтської частини сайту	18
1.2.4 Фреймворк Laravel для серверної частини сайту	20
1.2.5 CSS-фреймворк Tailwind CSS.....	21
1.2.6 Система контролю версій Git	22
1.3 Створення проекту і опис програмної реалізації.....	23
1.3.1 Формування вимог до веб-орієнтованої системи	23
1.3.2 Порядок розробки веб-сайту.....	26
1.3.2 Інформаційна структура сайту	27
1.3.3 Прототипування. Налаштування модульної сітки. Підбір кольорової палітри. Створення основних компонентів сайту	29
1.3.4 Створення логотипу.....	33
1.3.5 Створення дизайну сторінок. Методологія атомарного дизайну.....	35
1.3.6 Підготовка до розробки сайту	39
1.3.7 Файлова структура проекту	41
1.3.8 Проектування моделі бази даних	43
1.3.9 Розробка клієнтської частини.....	49
1.3.10 Розробка серверної частини.....	53

1.3.11 Тестування розробленого сайту	57
2 ЕКОНОМІЧНА ЧАСТИНА	60
3 ОХОРОНА ПРАЦІ	65
3.1 Аналіз небезпечних і шкідливих факторів, що впливають на працівника при розробці програмного комплексу	65
3.2 Гігієнічні вимоги до виробничого середовища.....	66
3.3 Пожежна безпека	69
ВИСНОВКИ	70
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	71
ДОДАТОК. Слайди мультимедійної презентації.....	73

ВСТУП

Стрімкий розвиток технологій зумовив революцію в освітній галузі, зокрема у процесі навчання та отриманні нових знань.

Онлайн-навчання стає дедалі актуальнішим, оскільки все більше людей віддають перевагу навчанню у власному темпі та у зручний для них час. У результаті виникла потреба в якісно розроблених платформах онлайн-навчання, які можуть запропонувати високоякісні освітні матеріали, щоб задовольнити зростаючий попит на онлайн-освіту.

Користувачі зможуть отримувати доступ до навчальних курсів з будь-якого пристрою, включаючи ноутбуки, планшети та мобільні телефони, без втрати функціональності та зручності використання. Крім того, розробка відкритої навчальної платформи розширить параметри навчання в онлайн-середовищі, дозволивши різноманітним користувачам брати участь у навчальному процесі, як початківцям так і досвідченим розробникам.

Актуальність теми дипломної роботи «Розробка та впровадження адаптивного веб-сайту онлайн навчання» обумовлена надзвичайною популярністю онлайн платформ навчання серед людства.

В рамках програмної реалізації дипломної роботи було поставлено завдання розробити адаптивний веб-сайт онлайн навчання на базі фреймворків Vue.js та Laravel.

Метою дипломної роботи є отримання готової програмної реалізації браузерної платформи онлайн навчання, з використанням сучасних технологій та інструментів розробки веб-додатків.

Відповідно до поставленої мети роботи визначено основні завдання дипломної роботи:

1. Провести аналіз предметної області.
2. Розглянути та проаналізувати існуючі аналогічні веб-додатки.
3. Розглянути особливості сучасних засобів по розробці web-додатків, виділити їх особливості, недоліки та переваги.

					<i>КГ 06. 21 000. 00 ДП ПЗ</i>	7
Зм.	Арк.	№ докум.	Підпис	Дата		

4. Розробити користувацький інтерфейс для web - сайту.
5. Розробити серверну частину web - сайту.
6. Виконати перевірку працездатності web - сайту.
7. Впровадити web-сайт у користування.

Об'єктом дослідження є проектування, розробка та впровадження адаптивного веб-сайту онлайн навчання.

Предметом дослідження є засоби та технології створення адаптивного веб-сайту онлайн навчання.

Для досягнення поставленої мети й виконання завдань застосовано сучасні технології та засоби розробки веб-застосунків, зокрема стек HTML/CSS/JS, фреймворк Vue.js, бібліотеки Vuex та InertiaJs, серверний фреймворк Laravel, CSS - фреймворк Tailwind CSS, редактор коду Microsoft Visual Studio Code, графічний редактор Figma, система контролю версій Git.

Результати дипломної роботи рекомендується використовувати для подальшого розвитку, додання та розширення функціоналу web-додатку.

					<i>КГ 06. 21 000. 00 ДП ПЗ</i>	
Зм.	Арк.	№ докум.	Підпис	Дата		8

1 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

1.1 Аналіз існуючих програмних рішень

У сучасну цифрову епоху веб-сайти та онлайн-платформи відіграють важливу роль у поширенні освітнього контенту та полегшенні засвоєння інформації. Для забезпечення успіху та ефективності веб-сайту, який розробляється, важливо провести детальний аналіз вже існуючих навчальних платформ.

Розглянемо три найвідоміші платформи для онлайн-навчання, а саме: ITVDN, Coursera та Prometheus, щоб отримати уявлення про їхні можливості, функціональність, користувацький досвід, а також основні переваги та недоліки. Цей аналіз допоможе визначити ключові аспекти, які потребують вдосконалення, виявити інноваційні функції та вдосконалити розробку власної платформи.

ITVDN - освітній online ресурс для IT-фахівців, створений у 2014 році. Мета проекту - навчання мовам програмування та інформаційним технологіям усіх, хто хоче стати професіоналом у сфері розроблення програмного забезпечення, проектування складних програмних систем, веб-розробки та в суміжних галузях. Навчання на ITVDN проводиться у формі окремих курсів, що дає змогу проходити навчання послідовно з нуля або вибірково за окремими технологіями для підвищення рівня кваліфікації фахівців [1].

					<i>КГ 06. 21 000. 00 ДП ПЗ</i>	9
Зм.	Арк.	№ докум.	Підпис	Дата		

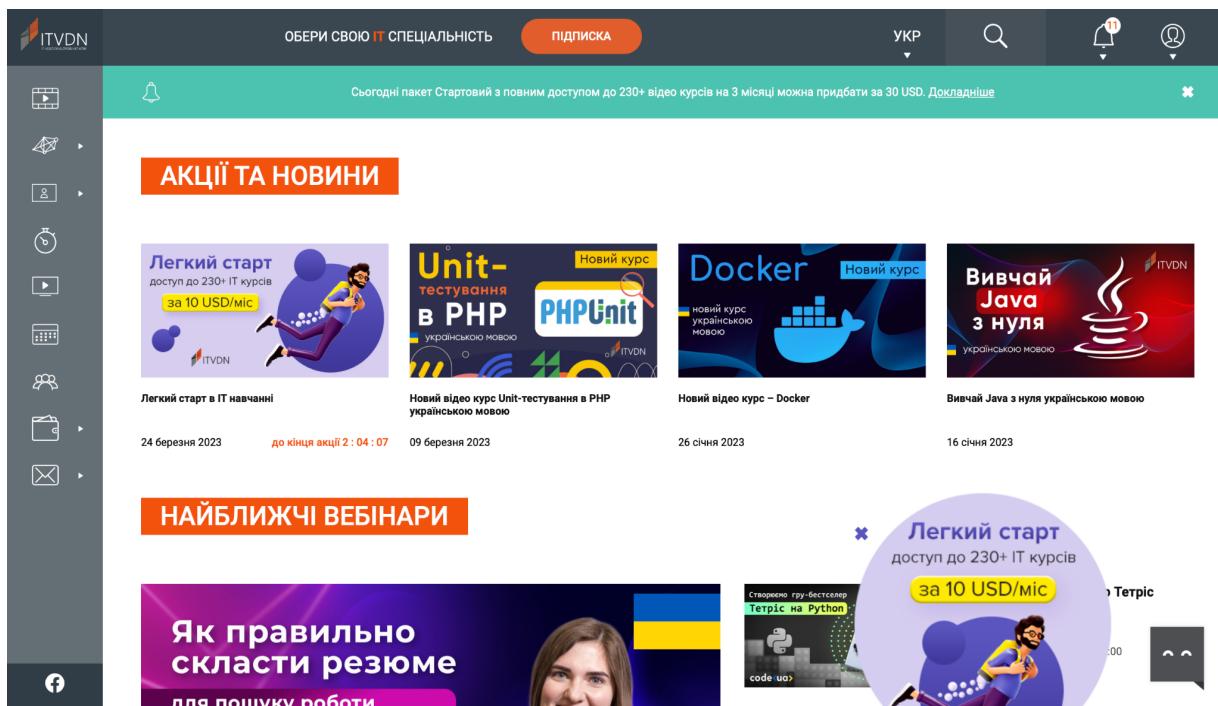


Рисунок 1.1 Інтерфейс сайту відеокурсів ITVDN

Prometheus - це веб-сайт для онлайн-навчання, який здобув визнання завдяки своїй спеціалізації на технічних курсах. Він забезпечує унікальний навчальний досвід, акцентуючи увагу на практичному навчанні та досвіді роботи над реальними проектами.

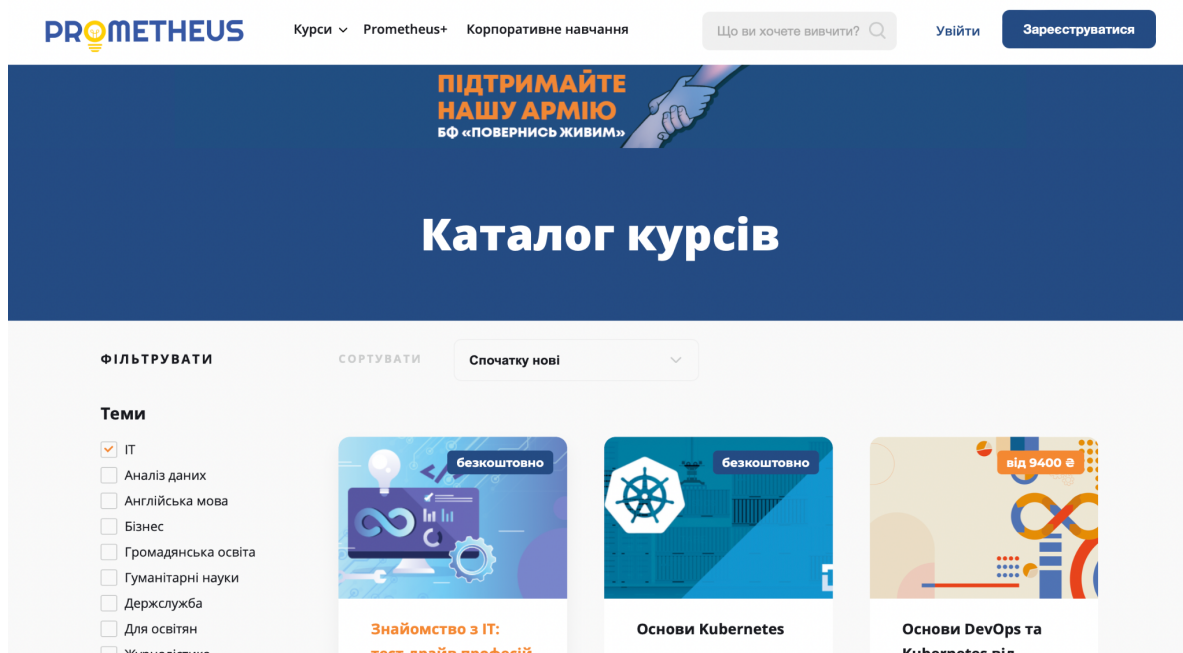


Рисунок 1.2 Інтерфейс платформи онлайн-навчання Prometheus

Зм.	Арк.	№ докум.	Підпис	Дата

КГ 06. 21 000. 00 ДП ПЗ

Coursera - один з найпопулярніших сайтів з онлайн-курсами з програмування та інших наук. Він пропонує велику кількість курсів, які охоплюють різноманітні теми, такі як розробка програмного забезпечення, аналіз даних, машинне навчання тощо.

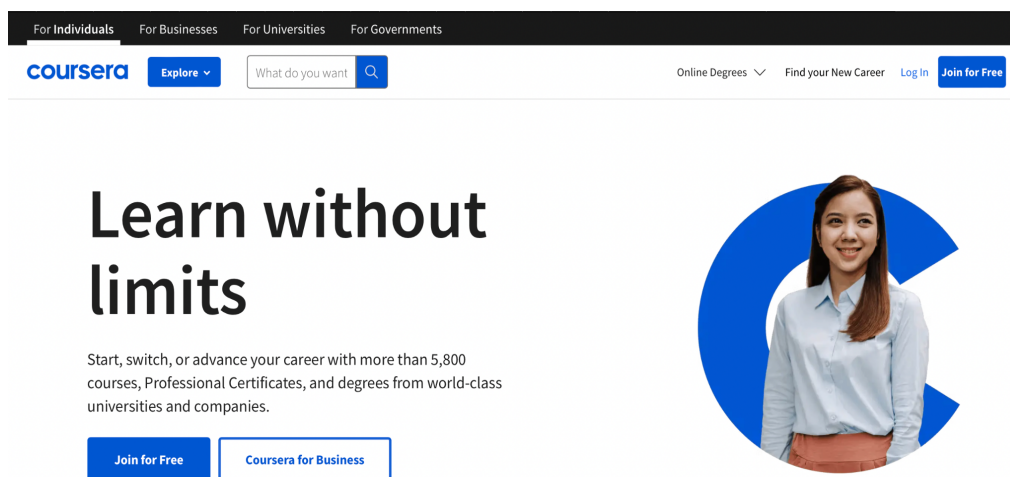


Рисунок 1.3 Інтерфейс платформи онлайн-навчання Coursera

Розглянемо основні переваги та недоліки даних веб-сайтів (табл. 1.1):

Таблиця 1.1 - Переваги та недоліки існуючих програмних рішень

Назва веб-ресурсу	Переваги	Недоліки
ITVDN	<ol style="list-style-type: none"> 1. Розмаїття курсів: ITVDN пропонує різноманітні курси з програмування, тестування, баз даних тощо. 2. Професійний рівень курсів: курси на ITVDN 	<ol style="list-style-type: none"> 1. Перевантаження: Деякі сторінки сайту можуть бути перевантажені великою кількістю елементів, що може вплинути на швидкість завантаження сторінки та

	<p>розробляються спеціалістами з великим досвідом роботи у своїй галузі, що дозволяє користувачам здобувати якісні знання та навички.</p> <p>3. Фільтри пошуку: на сайті є обширні параметри фільтрації курсів за різними критеріями, що дозволяє користувачам швидко знайти потрібний їм курс.</p>	<p>загальний користувацький досвід.</p> <p>2. Неінтуїтивність: деякі функції на сайті Coursera можуть бути складними для розуміння деякими користувачами, що може вплинути на зручність користування сайтом.</p> <p>3. Недостатній рівень безкоштовних курсів: ITVDN має деякі безкоштовні курси, але їхній рівень та обсяг може бути обмеженим, що може не задовольняти потреби деяких користувачів.</p>
Prometheus	<p>1. Великий вибір курсів: на сайті Prometheus є великий вибір курсів з програмування, дизайну, маркетингу та інших галузей.</p> <p>2. Висококваліфіковані викладачі: Викладачі Prometheus є досвідченими професіоналами у своїй</p>	<p>1. Висока вартість: курси на Prometheus є досить дорогими, що може бути недосяжним для деяких користувачів.</p> <p>2. Обмежений безкоштовний доступ: на Prometheus є деякі безкоштовні курси, проте їх обсяг та рівень можуть бути обмеженими, що може не задовольнити потреби деяких користувачів.</p>

	<p>галузі та мають великий досвід викладання.</p> <p>3. Гнучкість: курси на Prometheus можна проходити в будь-який зручний для користувача час і темп.</p>	
Coursera	<p>1. Велика кількість курсів: Coursera має вражаючий асортимент курсів, який охоплює більше ніж 4000 курсів з різних наукових галузей.</p> <p>2. Безкоштовні курси: на сайті Coursera є багато безкоштовних курсів.</p> <p>3. Фільтри пошуку: сайт Coursera має широкі можливості фільтрації курсів за різними критеріями.</p> <p>4. Простий та зрозумілий дизайн: сайт Coursera має простий та зрозумілий дизайн, що дозволяє користувачам легко</p>	<p>1. Велика кількість інформації: сайт Coursera містить велику кількість інформації, що може бути складною для сприйняття для деяких користувачів.</p> <p>2. Перевантаженість: деякі сторінки на сайті Coursera можуть бути перевантаженими багатьма елементами, що може впливати на швидкість завантаження сторінки та загальний комфорт користування сайтом.</p>

	зорієнтуватися на сайті та знайти необхідні курси.	
--	--	--

Таким чином, аналіз найпопулярніших платформ онлайн-навчання ITVDN, Coursera та Prometheus слугує орієнтиром для розуміння існуючих програмних рішень у сфері онлайн-навчання. Вивчивши їхні можливості та користувацький досвід, можна отримати важливу інформацію для подальшої розробки та вдосконалення власного сайту, а саме:

- веб-сайт повинен мати простий і зрозумілий дизайн, та не містити занадто багато інформації.
- на сайті повинна функціонувати пошукова система для забезпечення комфортного користування.
- великою перевагою буде наявність гнучкого графіку проходження курсів, без часових обмежень.
- кнопки та навігація повинні бути максимально зрозумілими кожному.

1.2 Вибір програмного забезпечення для реалізації проекту інформаційної системи

1.2.1 Середовище розробки Visual Studio Code

Visual Studio Code - це редактор коду, розроблений компанією Microsoft для Windows, Linux та macOS. Це потужний і зручний редактор вихідного коду, який можна використовувати для перегляду, редагування, запуску та налагодження програмного коду додатків. В основному його використовують фронтенд-розробники.

					<i>КГ 06. 21 000. 00 ДП ПЗ</i>	14
Зм.	Арк.	№ докум.	Підпис	Дата		

Однією з ключових особливостей Visual Studio Code є його чудова підтримка налагодження. Вбудований у VS Code дебаггер допомагає пришвидшити процес редагування, компіляції та налагодження коду.

- Можливість редагування різних файлів паралельно.

Іноді нам потрібно звернутися до коду одного файлу для іншого файлу. VS Code полегшує цю задачу, дозволяючи відкривати редактор поруч з одним і тим же файлом або різними файлами.

- Деякі можливості керування кодом.

Vs code надає нам мовні сервіси, такі як Peek Definition, Go to Definition, Find all References та Rename Symbol. Ці функції дуже корисні для кожного розробника [2].

1.2.2 Інструмент для створення інтерфейсу веб сайту Figma

Figma - це браузерний інструмент для спільного проектування користувацьких інтерфейсів, який дозволяє спільно працювати над створенням яскравих та інтерактивних прототипів. З моменту свого випуску в 2016 році Figma стала популярним інструментом як в індустрії веб-дизайну, так і в онлайн-спільнотах. Користувачі можуть співпрацювати та ділитися шаблонами, дизайнами та віджетами з мільйонами користувачів по всьому світу.

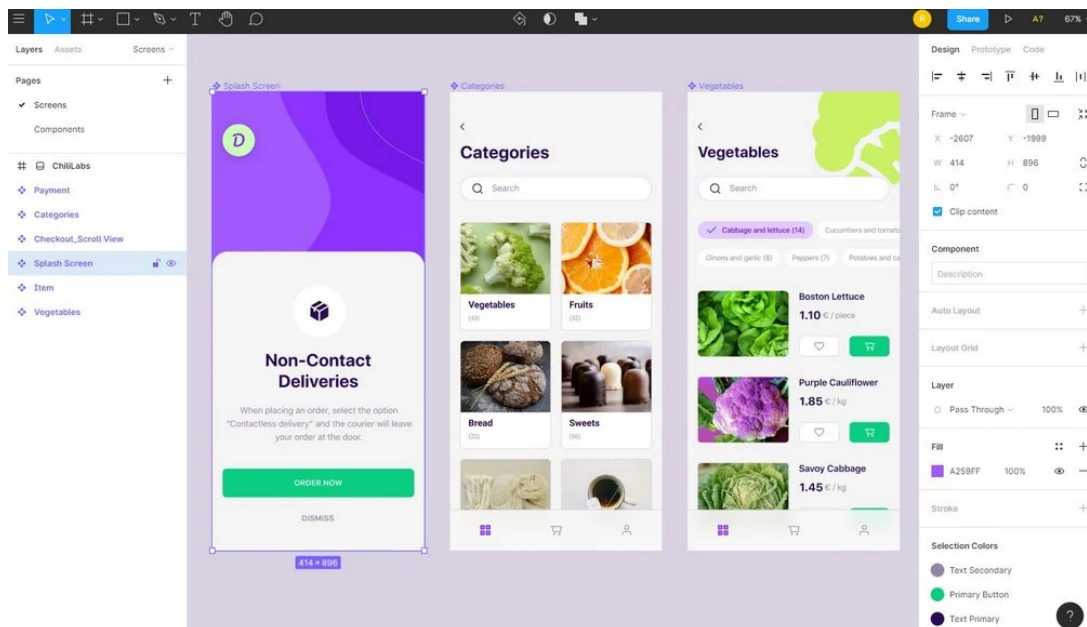


Рисунок 1.5 Інтерфейс Figma

Цей інструмент має наступні переваги:

- простий і зрозумілий у використанні
- Figma - це яскрава спільнота дизайнерів

Figma створила спільноту для своїх користувачів, де вони можуть ділитися шаблонами, плагінами та файлами. Системи дизайну, ілюстрації, іконки та багато іншого - все це можна використовувати.

- Figma має вбудований Slack

Завдяки інтеграції Figma зі Slack ви можете отримувати сповіщення про всі коментарі у Figma безпосередньо у Slack.

- FigJam - найкраща онлайн-дошка

FigJam - це онлайн-дошка. Користувачі можуть проводити мозковий штурм ідей, розробляти робочі процеси та планувати дизайн-спринти одночасно. Її можна використовувати безпосередньо зі стейкхолдерами для стартових зустрічей

- Figma має можливості створення прототипів та анімацій
- Figma є веб-орієнтованою [3].

Зм.	Арк.	№ докум.	Підпис	Дата

КГ 06. 21 000. 00 ДП ПЗ

1.2.3 Фреймворк Vue.js для клієнтської частини сайту

Фреймворк JavaScript - це сукупність вбудованих бібліотек JavaScript-коду, які надають веб-розробнику заздалегідь розроблений код для виконання рутинних завдань програмування. Фреймворки - це конструкції з певним контекстом, які допомагають створювати веб-додатки.

Сучасні JavaScript-фреймворки використовують шаблон розробки програмного забезпечення під назвою Model-View-View-Model. Він зазвичай використовується для розробки користувацьких інтерфейсів, які поділяють пов'язану логіку програмування на три взаємопов'язані елементи [4].

VueJS - це прогресивний JavaScript-фреймворк з відкритим вихідним кодом, який використовується для розробки інтерактивних веб-інтерфейсів.

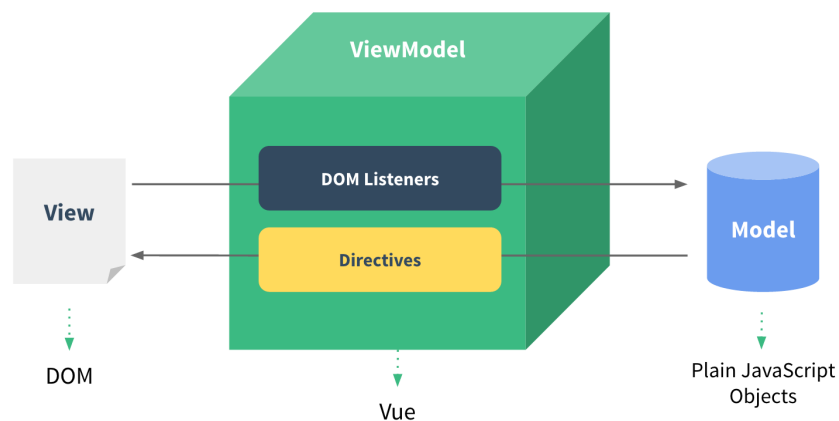


Рисунок 1.6 Структура фреймворку VueJS

Розглянемо переваги даного фреймворку:

- Простота

Достатньо написати 5-6 рядків коду, щоб створити додаток Vue.

Він усуває всі недоліки Angular, забезпечуючи легку, спрощену структуру для розробки веб-додатків і веб-сайтів з фронтенду.

- Архітектура на основі компонентів

Зм.	Арк.	№ докум.	Підпис	Дата

Як і Angular та React, фреймворк Vue.js використовує архітектуру на основі компонентів. Це означає, що весь код інтерфейсу програми можна розділити на незалежні компоненти.

- Багаторазове використання

Завдяки компонентному підходу Vue дозволяє створювати цілісні однофайлові компоненти для багаторазового використання. У середині компонента шаблон, логіка та стилі нерозривно пов'язані між собою.

- Висока продуктивність

Фреймворк Vue js надзвичайно легкий, він важить близько 20 КБ. Але це не впливає на продуктивність або продуктивність. Це один з найшвидших фреймворків для створення веб-інтерфейсів.

- Фреймворки та бібліотеки

Vue має велику кількість зовнішніх бібліотек та фреймворків для зручності розробників. Деякі популярні з них:

Nuxt.js - фреймворк для Vue, призначений для розробки продуктивних веб-додатків.

Bootstrap-Vue - це плагін для розробки додатків Vue з використанням компонентів Bootstrap.

- Virtual-DOM

Vue js використовує віртуальне представлення фактичного DOM веб-сторінки, створене за допомогою об'єктів Javascript.

- Реактивне двостороннє зв'язування даних

Це найважливіша особливість Vue.js. Завдяки Virtual DOM та функції двостороннього зв'язування, Vue.js забезпечує мінімальне використання коду для розробників.

- Декларативний рендеринг

Ядро системи Vue.js дозволяє декларативно рендерити дані в DOM. Використовуючи простий синтаксис шаблонів, Vue спрощує процес рендерингу веб-сторінок [5].

1.2.4 Фреймворк Laravel для серверної частини сайту

Laravel - це PHP-фреймворк з відкритим вихідним кодом, який є потужним і доступним для розуміння. Він слідує шаблону проектування "model-view-controller".

Laravel пропонує наступні ключові можливості:

- Модульність

Laravel надає 20 вбудованих бібліотек і модулів, які допомагають розширити можливості програми.

- Зручність у тестуванні

Laravel містить функції та помічники, які допомагають у тестуванні за допомогою різних тестових кейсів.

- Навігація

Laravel надає користувачеві гнучкий підхід до завдання маршрутів у веб-додатку.

- Управління конфігурацією

Веб-додаток, розроблений на Laravel, буде працювати в різних середовищах, а це означає, що його конфігурація буде постійно модифікуватися. Laravel надає узгоджений підхід для ефективного управління конфігурацією.

- Конструктор запитів та ORM

Laravel містить конструктор запитів, який допомагає створювати запити до баз даних за допомогою різноманітних простих ланцюжкових методів. Він надає ORM (Object Relational Mapper) та реалізацію ActiveRecord під назвою Eloquent.

- Шаблонний механізм

Laravel використовує Blade Template engine, легку мову шаблонів, яка використовується для створення ієрархічних блоків і макетів із заздалегідь визначеними компонентами, що включають динамічний контент.

- Автентифікація

Laravel полегшує розробку автентифікації, оскільки включає такі функції, як реєстрація, відновлення пароля та надсилання нагадувань про пароль.

- Redis

Laravel використовує Redis для підключення до існуючого сеансу та загального кешу. Redis взаємодіє з сесією напряму.

- Черги

Laravel включає в себе сервіси черг, такі як розсилка мейлів великій кількості користувачів або виконання певного завдання.

- Шина подій та команд

Laravel 5.1 включає командну шину, яка допомагає виконувати команди та диспетчеризувати події у простий спосіб [6].

1.2.5 CSS-фреймворк Tailwind CSS

CSS розшифровується як Cascading Style Sheets. CSS відповідає за оформлення і зовнішній вигляд веб-сторінок.

Tailwind CSS - це універсальний CSS-фреймворк, розроблений для того, щоб користувачі могли створювати веб додатки швидше і зручніше. Завдяки використанню спеціальних класів ви можете керувати оформленням, кольором, відступами, шрифтами, тінями тощо, щоб створити повністю кастомний дизайн компонентів - не залишаючи HTML і не пишучи жодного рядка кастомного коду CSS.

```
<div class="p-6 max-w-sm mx-auto bg-white rounded-xl shadow-lg flex items-center space-between">
  <div class="shrink-0">
    
  </div>
  <div>
    <div class="text-xl font-medium text-black">ChitChat</div>
    <p class="text-slate-500">You have a new message!</p>
  </div>
</div>
```

Рисунок 1.7 Приклад використання Tailwind CSS

Наведемо кілька переваг використання даного фреймворку:

- Зменшення кількості кастомного коду CSS.

За допомогою Tailwind ви можете стилізувати елементи, застосовуючи існуючі класи безпосередньо до HTML.

- Розмір файлів CSS значно скорочується.

Використовуючи такі утиліти, як flexbox і padding в Tailwind, більшість стилів можна використовувати багаторазово, тому вам рідко потрібно писати новий CSS.

- Немає потреби вигадувати назви класів.

У Tailwind ви обираєте класи з попередньо визначеної системи дизайну.

- Ви можете вносити зміни більш безпечно.

На відміну від CSS, утилітарні класи в вашому HTML є локальними. Це означає, що ви можете змінювати їх, не турбуючись про те, що зламаєте щось інше на вашому сайті [7].

1.2.6 Система контролю версій Git

Git - це відкрита розподілена система контролю версій з відкритим вихідним кодом. Він розроблений для координації роботи між розробниками. Контроль версій дозволяє відстежувати і працювати разом з членами команди в одному робочому середовищі.

Git має такі можливості:

- Відкритий вихідний код

Git - це інструмент з відкритим кодом.

- Можливість розширення

Git є розширюваним, а це означає, що коли кількість користувачів збільшується, Git може легко обробляти такі ситуації.

- Розподілений.

Одна з чудових речей в Git - це його розподіленість. Розподіленість означає, що замість того, щоб перемикати проект на іншу машину, можна просто створити "клон" всього сховища.

- Безпека

Git є безпечним. Він використовує SHA1 (Secure Hash Function) для іменування та ідентифікації об'єктів у своєму сховищі.

- Швидкість

Git дуже швидкий, завдяки чому він може виконати всі завдання за короткий проміжок часу.

- Підтримує нелінійну розробку

Git підтримує плавне розгалуження та злиття, що допомагає візуалізувати та орієнтуватися в нелінійній розробці.

- Розгалуження та злиття

Розгалуження та злиття - чудові можливості Git, які відрізняють його від інших SCM-інструментів. Git дозволяє створювати декілька гілок, які не впливають одна на одну. Можна виконувати такі операції, як створення, видалення та злиття гілок, і ці операції займають лише кілька секунд.

- Область підготовки (Staging Area)

Її можна розглядати як попередній перегляд нашого наступного коміту, а також як проміжну область, де коміти можна відформатувати і переглянути перед завершенням [8].

1.3 Створення проекту і опис програмної реалізації

1.3.1 Формування вимог до веб-орієнтованої системи

Проаналізувавши сайти конкурентів, було вирішено створити максимально мінімалістичну та зручну для користувача веб-систему. Було прийнято рішення не перевантажувати сайт великою кількістю додаткового функціоналу, зайвої інформації та рекламних банерів на сторінках. Призначення кожного функціонального елемента системи має бути зрозумілим кожному користувачеві Інтернету.

Веб-сайт повинен містити такі функціональні компоненти:

- форми реєстрації та авторизації для користувачів;
- особистий кабінет користувача (після реєстрації на сайті);

- каталог курсів;
- пошук по сайту;
- інформація про кожен курс;
- форма зворотнього зв'язку.

Історія користувача (user story) - це короткий, простий опис функціоналу, який хотів би бачити користувач. Зазвичай це лише одне-два речення, написані за такою формулою:

Як [роль користувача], я хочу [функціональність], щоб [переваги].

Розроблюваний веб-сайт передбачатиме наявність трьох користувацьких ролей: “Гість”, “Користувач” та “Адміністратор”.

Гість - це особа, яка відвідує веб-сайт без облікового запису або облікових даних для входу в систему. Гості мають обмежений доступ до функцій і можливостей, доки не створять обліковий запис. Після створення облікового запису гість стає повноцінним зареєстрованим користувачем.

Адміністратор - це привілейований користувач, який має доступ до розширених функцій та елементів керування сайтом. Він відповідає за обслуговування та управління контентом веб-сайту.

Таблиця 1.2 - User story

№	Роль користувача	Формулювання користувацької вимоги
1	Гість, Користувач	Як гість/користувач, я хочу переглядати список доступних курсів на веб-сайті онлайн навчання, щоб визначити, чи варто мені реєструватися на сайті.
2	Гість, Користувач	Як гість/користувач, я хочу бачити короткий опис кожного курсу, щоб зрозуміти, що включається в цей курс і чи він мені підходить.

3	Гість, Користувач	Як гість/користувач, я хочу мати можливість знайти курс за ключовими словами або тегами, щоб знайти найбільш відповідний курс для своїх потреб.
4	Гість, Користувач	Як гість/користувач, я хочу мати можливість зв'язатися з підтримкою сайту через форму зворотного зв'язку, щоб отримати допомогу в разі виникнення проблем.
5	Гість, Користувач	Як гість/користувач, я хочу мати змогу змінювати кольорову тему сайту, щоб досвід користування був більш комфортним та персоналізованим для мене.
6	Користувач	Як користувач, я хочу мати можливість створити обліковий запис, щоб мати можливість зберегти вподобаний курс.
7	Користувач	Як користувач, я хочу мати можливість змінити свій пароль, щоб підвищити безпеку свого облікового запису.
8	Користувач	Як користувач, я хочу мати можливість відновлення пароля при логіні у разі загублення пароля, щоб отримати доступ до особистого кабінету.
9	Користувач	Як користувач, я хочу мати можливість додати курс до списку обраного, щоб я міг швидко знайти його пізніше.
10	Користувач	Як користувач, я хочу мати можливість змінити свої особисті дані (наприклад, електронну адресу або ім'я), щоб оновити свій профіль на сайті.
11	Адміністратор	Як адміністратор, я хочу мати можливість змінювати інформацію про курс, щоб оновлювати та покращувати інформацію для користувачів.

Зм.	Арк.	№ докум.	Підпис	Дата

КГ 06. 21 000. 00 ДП ПЗ

12	Адміністратор	Як адміністратор, я хочу мати можливість видаляти курси з сайту, щоб утримувати сайт актуальним.
13	Адміністратор	Як адміністратор, я хочу мати можливість додавати нові курси на сайт, щоб користувачі могли знайти більше матеріалів.
14	Адміністратор	Як адміністратор, я хочу мати можливість надавати технічну підтримку користувачам, щоб вони могли швидко та легко вирішувати проблеми, які виникають при використанні сайту.

1.3.2 Порядок розробки веб-сайту

Веб-розробка - це комплексний процес, що включає кілька напрямків та етапів реалізації. Основними етапами веб-розробки є аналіз, проектування, розробка, тестування та оцінка, впровадження та технічне обслуговування.



Рисунок 1.8 Етапи веб-розробки

1.3.2 Інформаційна структура сайту

Структура веб-сайту - це певна ієрархія, послідовність та порядок розташування сторінок веб-сайту, орієнтована на забезпечення зручного інтерфейсу для користувачів. Вона пов'язує всі ваші веб-сторінки за допомогою навігаційної панелі меню, внутрішніх посилань і наповнення.

Інформаційна система, що розробляється, включатиме низку веб-сторінок, оскільки потребуватиме надання досить великого обсягу контенту. Відповідно, навігація повинна бути максимально комфортною та інтуїтивно зрозумілою. Крім того, варто анімувати кнопки та статичні елементи, щоб зробити сайт візуально привабливим для користувачів.

Розглянемо основні складові розроблюваної інформаційної системи та їх призначення:

- головна - включає у себе пошук курсів за тегами та перегляд популярних курсів;
- про нас - сторінка, що включає у себе інформацію про веб-сайт;
- курси - сторінка з переліком доступних курсів;
- зв'язок - сторінка з контактною формою для зв'язку;
- особистий кабінет - сторінка для перегляду збережених курсів та відео, редагування особистої інформації, зміна паролю.

Для створення веб-сторінки можна використовувати три типи структури сайту: послідовну модель, ієрархічну структуру сайту та мережеву модель.

Лінійна структура сайту - це найпростіший спосіб упорядкування всієї інформації, яку потрібно запропонувати користувачам. При даній структурі сайту контент розміщується в послідовному порядку.

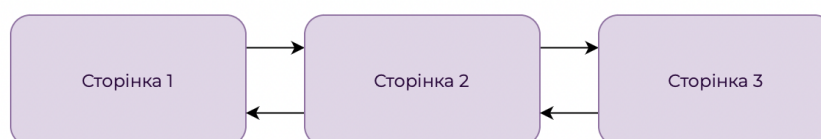


Рисунок 1.9 Лінійна структура веб-сайту

Ієрархічна модель використовується для впорядкування великих обсягів інформації. Тому розробники, які обирають такий тип структури веб-сайту, можуть запропонувати користувачам багато інформації і показувати її від загальної до більш детальної.

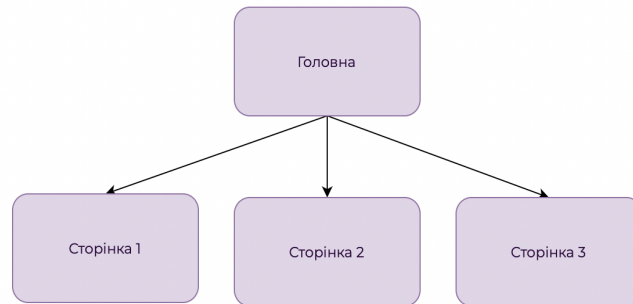


Рисунок 1.10 Ієрархічна структура веб-сайту

Насамкінець, мережева модель є однією з найбільш зручних в організації інформаційного простору. Ця модель передбачає вільну навігацію без встановлених шаблонів і віддає перевагу вільному переміщенню по сайту.

Для веб-сайту була обрана мережева структура, схема якої представлена на рисунку:

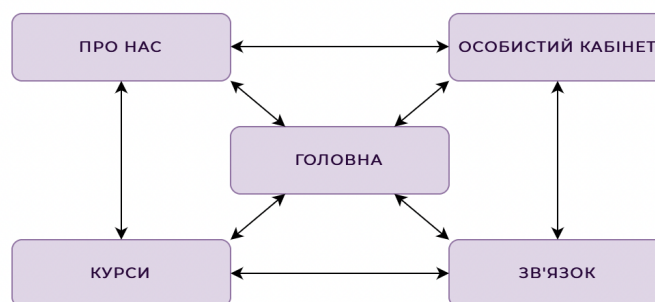


Рисунок 1.11 Мережева структура сайту

Особливістю мережевої структури є те, що не існує чіткого шаблону, якого слід дотримуватися, коли користувач заходить на веб-сайт (на відміну від лінійної та ієрархічної структури).

1.3.3 Прототипування. Налаштування модульної сітки. Підбір кольорової палітри. Створення основних компонентів сайту

Проектування дизайну будь-якого веб-сайту розпочинається зі створення прототипу. Прототип - це ескізна версія певного додатку, яка дозволяє дослідити власні ідеї та продемонструвати користувачам задум функціональності або загальну концепцію дизайну.



Рисунок 1.12 Приклад прототипу веб-сайту

Модульна сітка - це система упорядкування елементів у макеті, заснована на колонках, рядках і відступах між ними. За направляючими сітки дизайнер вирівнює всі елементи сайту: текстову інформацію, зображення, кнопки.

Система модульної сітки з 12 колонками однакової ширини забезпечує високу надійність, гнучкість і надає безліч варіантів організації структури, таким чином, дизайн буде створено на основі таких параметрів:

Кількість колонок - 12;
Ширина колонок - 98 рх;
Проміжок між колонками- 20 рх;
Поля - 20 рх.

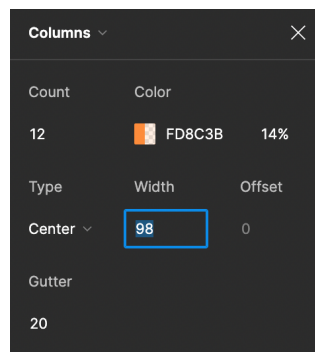


Рисунок 1.13 Налаштування модульної сітки

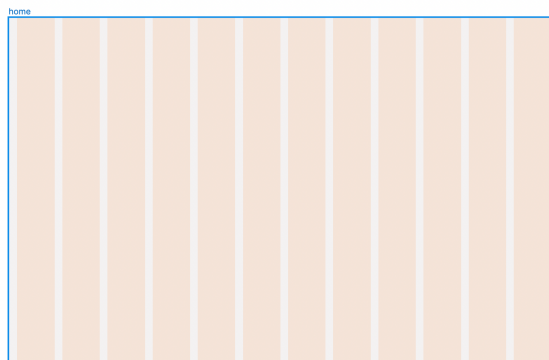


Рисунок 1.14 Модульна сітка

Підбір колірної схеми сайту

Колірна схема сайту - це набір кольорів, які дизайнер обирає для дизайну сайту. Вибір правильної кольорової палітри має велике значення при створенні веб-сайту, оскільки вона задає загальний стиль і створює відповідну атмосферу на сайті. Грамотно підібрані кольори покращують користувацький досвід, посилюють брендинг і надають сайту візуально привабливого вигляду.

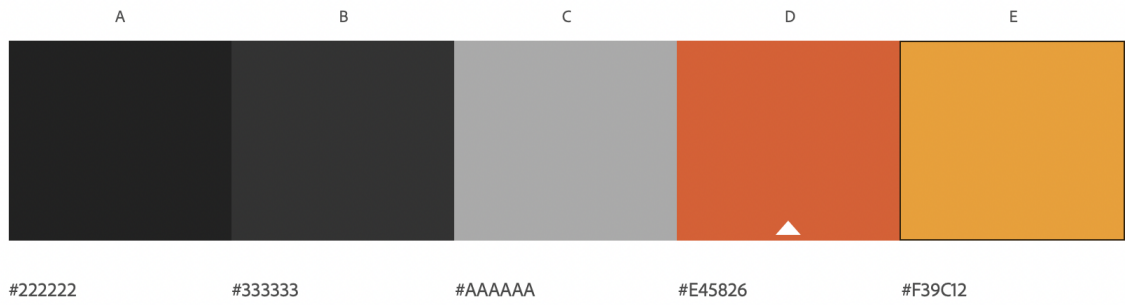


Рисунок 1.15 Колірна схема сайту

Акцентним кольором було обрано помаранчевий. Згідно з психологією кольору, помаранчевий - це невичерпна енергія в чистому вигляді. Цей колір також характеризується життєрадісністю, захопленістю, допомагає вирішувати складні завдання, дарує радість, забезпечує спонтанність дій.

Для тексту було обрано #AAAAAA та #FEFEFE.

Для заднього фону - #222222 та #333333.

Важливо переконатися, що кольорова палітра є безпечною для людей з дальтонізмом, тому перед використанням я перевіряю її за допомогою спеціального інструменту Color Blind Safe на сайті Adobe Color. Згідно з результатами тестування, палітра повністю безпечна для людей з дальтонізмом (див. рис.).



Рисунок 1.16 Перевірка кольорів

Однією з основних характеристик при виборі кольорів є контраст тексту на фоні, для цього я перевіряю контрастність за допомогою інструменту Contrast Checker:

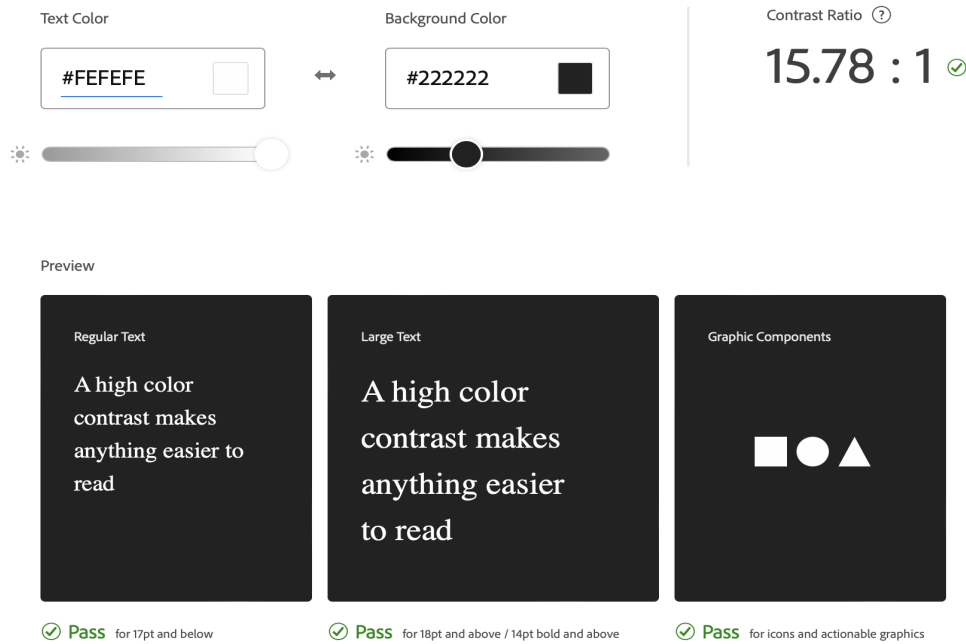


Рисунок 1.17 Перевірка на контрастність

Створення компонентів - header, sidebar, footer.

Саме компоненти веб-сторінки у веб-дизайні займають центральне місце у формуванні цілісного та зручного користувацького інтерфейсу.

Базова структура сторінки буде складатися з чотирьох частин: верхня панель, бічна панель навігації, нижня панель та контент.

Верхня панель - це верхня частина веб-сторінки, яка є важливим елементом дизайну, оскільки, коли користувачі починають взаємодіяти з сайтом, це, як правило, перше, що вони бачать на ньому.



Рисунок 1.18 Верхня панель

Майбутній сайт міститиме навігаційну бічну панель, що дозволить користувачам швидко знаходити те, що їм потрібно. Крім того, на бічній панелі відобразатиметься аватар та ім'я користувача, а також кнопка для переходу до особистого кабінету. Забезпечення зручного доступу до найбільш значущих сторінок сайту буде одним з основних призначень бічної панелі.

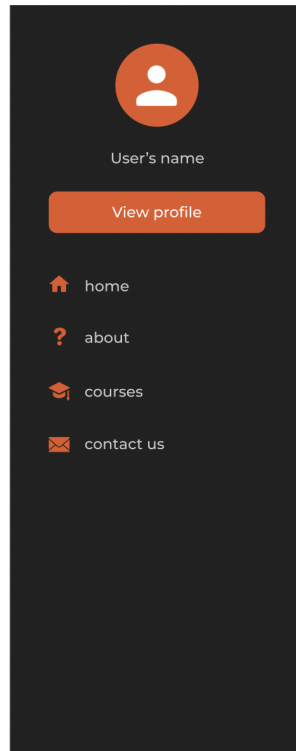


Рисунок 1.19 Бічна панель

1.3.4 Створення логотипу

Під терміном "логотип" зазвичай розуміють усі знаки, котрі репрезентують бренд. Якісно розроблений логотип формує довіру, підтверджуючи ваш професіоналізм, і спонукає людей залишатися з вами. Досить часто форма дизайну логотипу співвідноситься з тематикою компанії або має відношення до неї і включає в себе основні елементи бренду. Колір повинен бути сміливим і привертати увагу, але не перевантажуючим або незручним.

Розрізняють три типи логотипів:

- Wordmark (текстовий логотип)
- Logotype

- Brandmark (торгова марка)

Вордмарк (wordmark) - це текстовий логотип, який є найпростішим з точки зору дизайну.



Рисунок 1.20 Приклади текстового логотипу

Логотип (logotype) дуже схожий на "Wordmark", оскільки вони обидва є переважно текстовими логотипами. Логотип відрізняється тим, що він включає символіку в текстовий дизайн. Це досягається перетворенням однієї або декількох літер на іконку, яка символізує логотип.



Рисунок 1.21 Приклад логотипу

Брендмарка - це розпізнавальна іконка, яка представляє бренд і здебільшого рідко містить текст. Це найуніверсальніша частина дизайну логотипу.



Рисунок 1.22 Приклад брендмарки

Почнемо з розробки брендмарки сайту, тобто ідентифікаційної іконки. У програмі Figma я використовую інструмент перо, щоб намалювати контур майбутньої іконки.

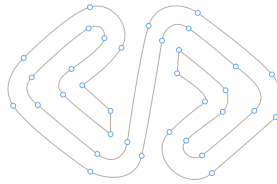


Рисунок 1.23 Контур брендмарки сайту

Потім вношу точкові корективи та змінюю колір і ширину.

У результаті я отримую дві версії іконки - контурну і зафарбовану.



Рисунок 1.24 Брендмарка сайту у двох варіантах

Імпортую їх у формат PNG для подальшої роботи.

1.3.5 Створення дизайну сторінок. Методологія атомарного дизайну

Атомарний дизайн - це методологія, котра складається з п'яти окремих етапів, що сприяють створенню систем розробки інтерфейсів у більш продуманий та ієрархічний спосіб. Існує п'ять стадій атомарного дизайну:

1. Атоми
2. Молекули
3. Організми
4. Шаблони
5. Сторінки

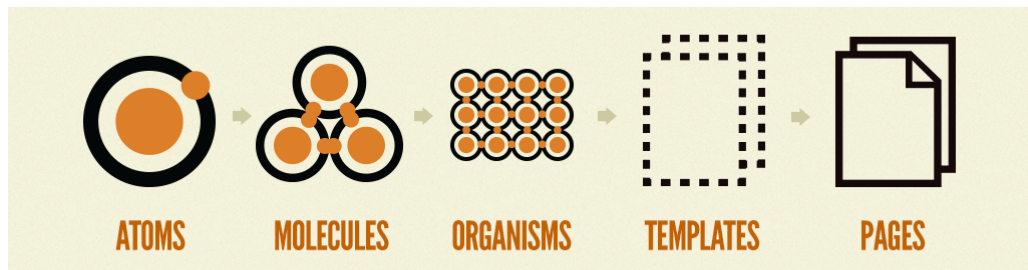


Рисунок 1.25 Стадії атомарного дизайну

Атоми інтерфейсів слугують фундаментальними будівельними блоками, з яких складаються всі наші користувацькі інтерфейси. До складу атомів входять базові елементи HTML, такі як поля вводу, кнопки та інші, які не можуть бути розбиті на частини без втрати функціональності.

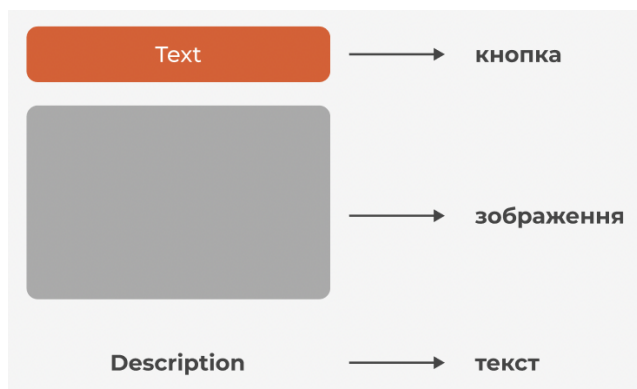


Рисунок 1.26 Приклад атомів розробляємого інтерфейсу

Молекули - це відносно прості сукупності елементів інтерфейсу, що функціонують разом як єдине ціле. Поєднуючись, абстрактні атоми несподівано отримують призначення. Атом тексту тепер визначає опис. Натискання на кнопку атома тепер служить переходом на іншу сторінку. Результатом є простий, портативний, багаторазовий компонент.

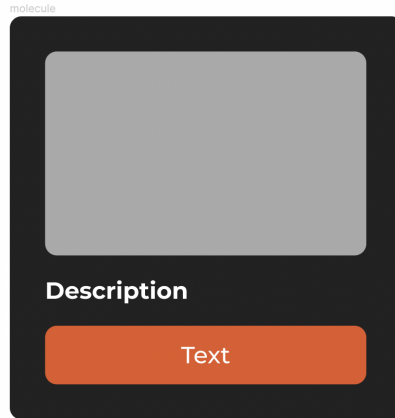


Рисунок 1.27 Приклад молекули розробляемого інтерфейсу

Організми - це досить великі компоненти інтерфейсу, що складаються з груп молекул та/або атомів, та/або інших організмів. Ці організми формують окремі секції інтерфейсу.

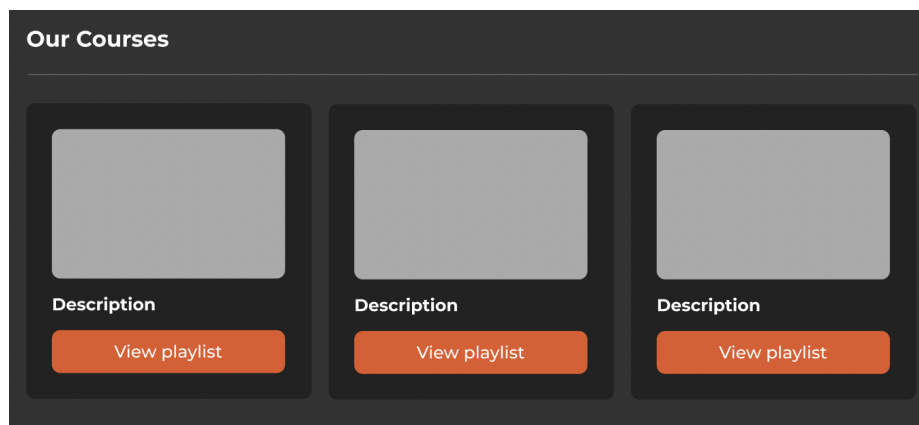


Рисунок 1.28 Приклад організму розробляемого інтерфейсу

Шаблони - це об'єкти на рівні сторінки, які розміщують компоненти в макеті та формулюють основну структуру контенту дизайну.

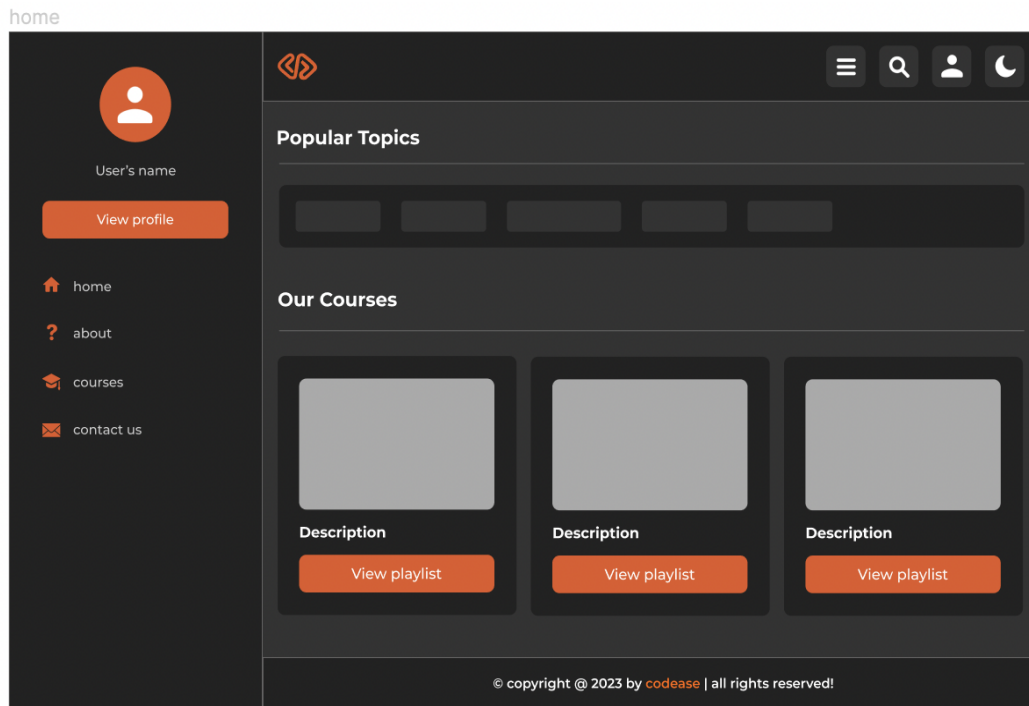


Рисунок 1.29 Шаблон сторінки Головна

Даний шаблон домашньої сторінки відображає всі необхідні компоненти сторінки, які функціонують разом, що забезпечує контекст для цих відносно абстрактних молекул і організмів. При створенні ефективної системи дизайну дуже важливо продемонструвати, як компоненти виглядають і функціонують разом у контексті макета, щоб довести, що елементи утворюють гармонійно функціонуюче ціле.

Сторінки - це конкретні екземпляри шаблонів, які показують, як виглядає інтерфейс користувача з реальним репрезентативним контентом. Ґрунтуючись на попередньому прикладі, можна взяти шаблон домашньої сторінки і додати до нього репрезентативний текст, зображення та медіа, щоб показати реальний вміст у дії.

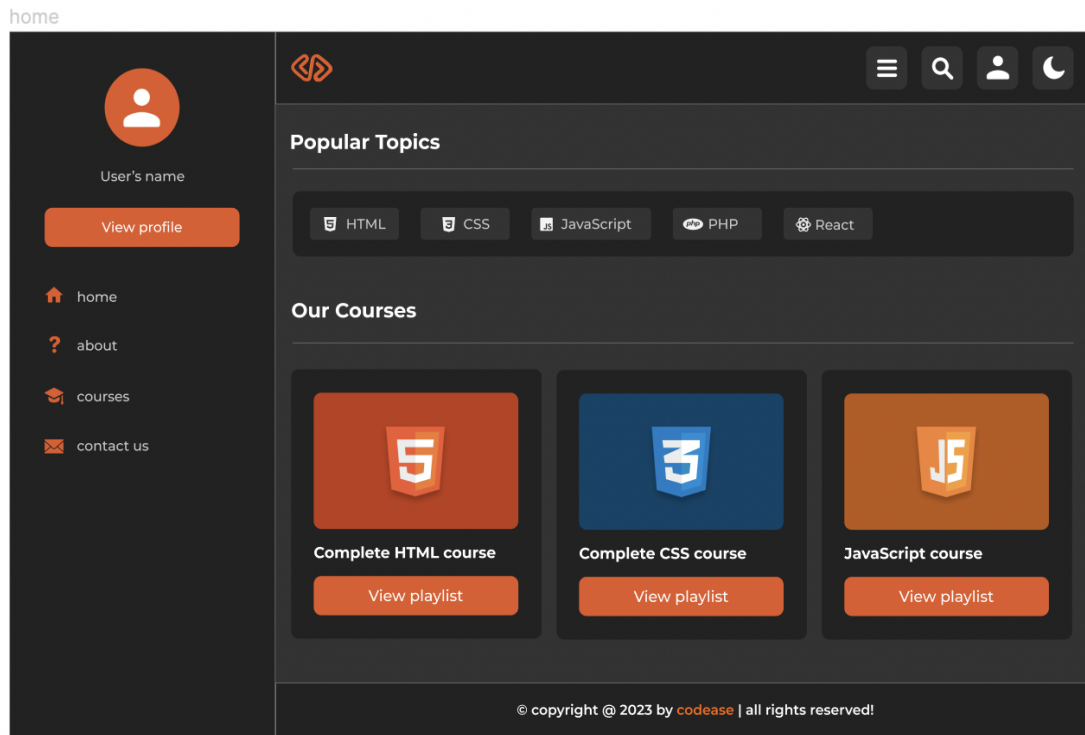


Рисунок 1.30 Дизайн сторінки “Головна”

Завдяки поєднанню цих п'яти окремих етапів створюються ефективні дизайни користувацьких інтерфейсів [9].

В результаті використання методології атомарного дизайну було розроблено дизайн основних сторінок майбутнього веб-сайту.

1.3.6 Підготовка до розробки сайту

Перш ніж приступити до створення веб-сайту на основі Vue.js і Laravel, необхідною умовою є створення відповідного робочого середовища, сприятливого для використання цих передових технологій.

Для початку роботи з Vue.js важливо мати встановлену останню версію Node.js, оскільки ця технологія є основою Vue.js. Далі фреймворк Vue.js встановлюємо за допомогою менеджера пакетів npm. Після встановлення Vue.js можна створити новий проект за допомогою шаблону Vue CLI. Для спрощення розробки також можна встановити додаток Vue.js devtools, розширення для браузера, яке полегшує налагодження веб-додатків.

					<i>КГ 06. 21 000. 00 ДП ПЗ</i>	39
Зм.	Арк.	№ докум.	Підпис	Дата		

Далі перейдемо до фреймворку Laravel. Готуючись до розробки, необхідно переконатися, що на вашому комп'ютері встановлено PHP. Згодом Laravel можна встановити через Composer, менеджер пакетів для PHP. Після успішної інсталяції, створення нового проекту можна здійснити за допомогою команди Artisan. Для оптимізації процесу розробки доцільно встановити додаток Laravel Debugbar, який полегшує моніторинг запитів веб-додатків і показників продуктивності.

Для ефективного використання Vue.js та Laravel дуже важливо встановити всі необхідні залежності (див. рис. 1.33). Ці залежності використовуються для покращення функціональності. Одні із них це vuex, axios, inertia js, tailwind css, vite. Менеджер пакетів npm є відповідним інструментом для встановлення залежностей.

```
"devDependencies": {
  "@fortawesome/fontawesome-free": "^6.4.0",
  "@inertiajs/vue3": "^1.0.0",
  "@tailwindcss/forms": "^0.5.3",
  "@vitejs/plugin-vue": "^4.0.0",
  "autoprefixer": "^10.4.12",
  "axios": "^1.1.2",
  "laravel-vite-plugin": "^0.7.2",
  "postcss": "^8.4.18",
  "sass": "^1.60.0",
  "sass-loader": "^13.2.1",
  "tailwindcss": "^3.2.1",
  "vite": "^4.0.0",
  "vue": "^3.2.41",
  "vue-loader": "^17.1.0",
  "vue-router": "^4.1.6",
  "vue-template-compiler": "^2.7.14",
  "vuex": "^4.1.0"
}

"require": {
  "php": "^8.1",
  "guzzlehttp/guzzle": "^7.2",
  "inertiajs/inertia-laravel": "^0.6.8",
  "laravel/framework": "^10.0",
  "laravel/sanctum": "^3.2",
  "laravel/tinker": "^2.8",
  "tightenco/ziggy": "^1.0"
}
```

Рисунок 1.31 Залежності проекту

Inertia.js - це фреймворк, який дозволяє створювати унікальні веб-сайти з використанням серверного рендерингу. Цей інструмент поєднує в собі функціональність Vue.js на стороні клієнта та Laravel на стороні сервера.

Vueх - це шаблон управління станами для Vue.js, який дозволяє організувати та керувати станом веб-сайту на рівні компонентів.

Axios - це HTTP-клієнт, призначений для відправки запитів з Vue.js до сервера. Він спрощує взаємодію з АРІ та обробку відповідей.

Vite.js дозволяє розробникам налаштувати середовище розробки для таких фреймворків, як Vue і React.

1.3.7 Файлова структура проекту

Дипломний проект містить наступні директорії:

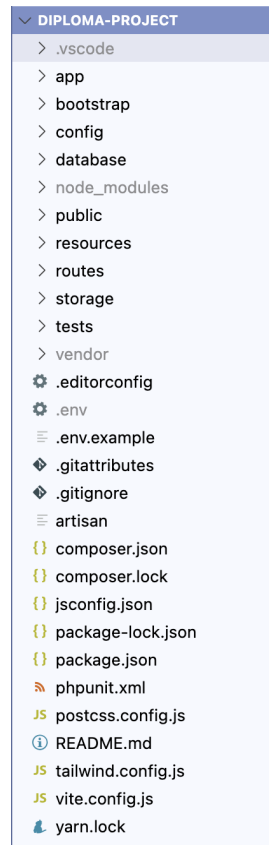


Рисунок 1.32 Файлова структура проекту

- Директорія App

Директорія App містить основний код проекту.

- Директорія Bootstrap

Каталог bootstrap містить файл app.php, який завантажує фреймворк. У цьому каталозі також знаходиться каталог кешу, який містить згенеровані фреймворком файли для оптимізації продуктивності, такі як кеш-файли маршрутів і сервісів.

- Директорія Config

Каталог конфігурації, як випливає з назви, містить усі конфігураційні файли програми.

- Директорія Database

Каталог бази даних містить міграції баз даних, моделі та зразки.

- Директорія Public

Публічний каталог містить файл index.php, який є точкою входу для всіх запитів, що надходять до сайту, і налаштовує автозавантаження. У цьому каталозі також зберігаються ресурси, такі як зображення та стилі.

- Директорія Resources

Каталог ресурсів містить елементи інтерфейсу, а також початкові, не скомпільовані ресурси, такі як CSS або JavaScript.

- Директорія Routes

Каталог маршрутів містить усі маршрути, створені для проекту. За замовчуванням до складу Laravel входить декілька файлів маршрутів: web.php, api.php, console.php та channels.php.

Основна частина проекту міститься у каталозі app. У каталозі app міститься низка допоміжних каталогів, таких як Console, Http та Providers. Каталоги Console і Http можна розглядати як інтерфейс API до ядра програми. Протокол HTTP і CLI є механізмами для взаємодії з вашим додатком, але вони фактично не містять логіки програми. Іншими словами, це два способи передачі команд вашій програмі. Каталог Console містить всі ваші команди Artisan, в той час як каталог Http містить ваші контролери, проміжне програмне забезпечення і запити [10].

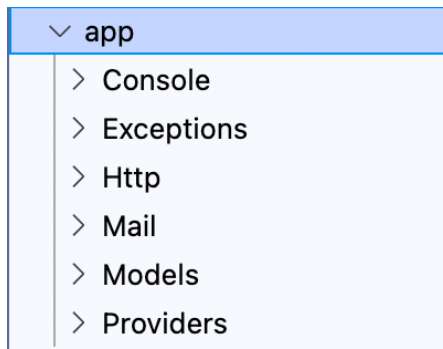


Рисунок 1.33 Директорія app

1.3.8 Проектування моделі бази даних

За основу проектування бази даних була взята модель діаграми "сутність-зв'язок" (ER), яка допомагає наочно відобразити взаємозв'язки між сутностями, представленими в базі даних. Реляційна модель сутностей - це модель для ідентифікації сутностей, які повинні бути представлені в базі даних, і відображення того, як ці сутності пов'язані між собою. ER-модель складається з сутностей, атрибутів та зв'язків між сутностями в системі баз даних [11].

ER-діаграма бази даних зображена на рисунку 1.36.

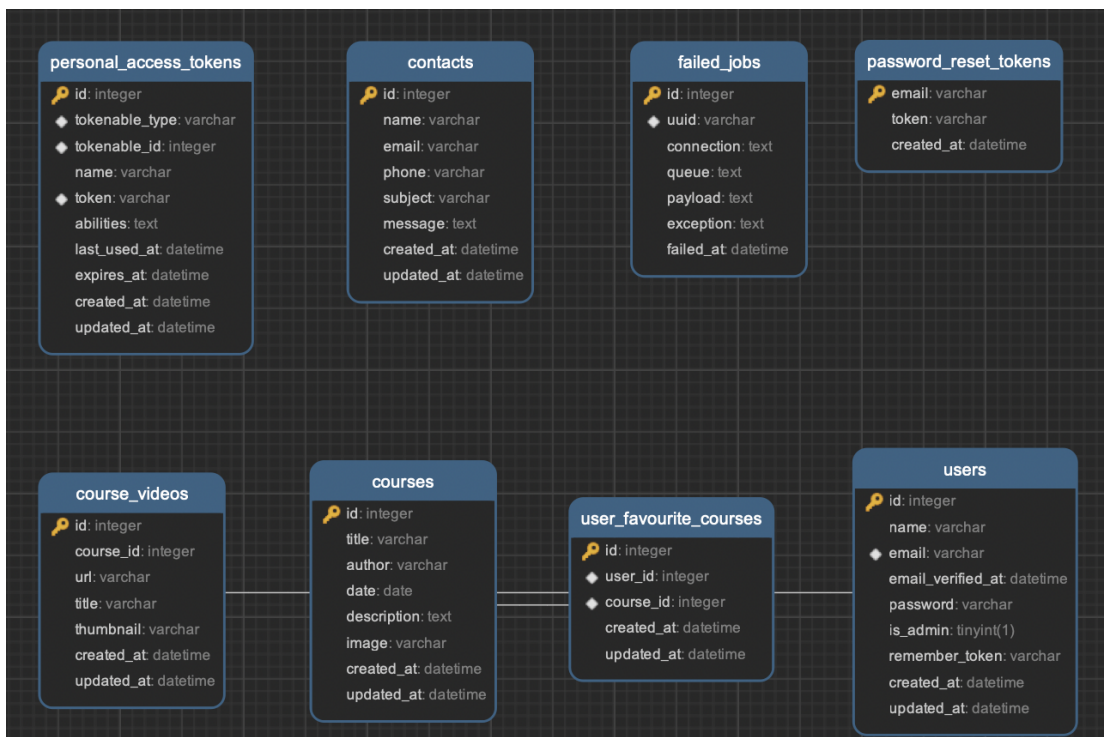


Рисунок 1.34 ER-діаграма бази даних

Отже, для забезпечення функціональності системи було розроблено базу даних, яка налічує вісім сутностей

- users - інформація про користувачів;
- contacts - використовується для зберігання звернень отриманих через контактну форму;
- courses - використовується для зберігання даних про курси;
- course_videos - використовується для зберігання даних про відео що знаходяться у курсі;
- user_favourite-courses - інформація про збережені користувачем курси;
- failed_jobs - використовується для зберігання інформації про невдалі завдання в системі черг Laravel;
- personal_access_token - інформація про персональні токени доступу;
- password_reset_tokens - використовується для зберігання токенів відновлення пароля;

Кожна з наведених вище сутностей містить атрибути, що наведені у Таблиці

1.3

Таблиця 1.3 - Опис таблиць бази даних

	Поле	Зміст	Тип	Ключі	Обмеження
users	id	Ідентифікатор	integer	РК	Не пустий
	name	Ім'я	varchar		Не пустий
	email	Пошта	varchar		Не пустий
	email_verified_at	Дата верифікації пошти	datetime		
	password	Пароль	varchar		Не пустий

	is_admin	Наявність прав адміністратора	tinyint(1)		Не пустий
	remember_token	Токен збереження сесії	varchar		
	created_at	Відмітка часу створення	datetime		
	updated_at	Відмітка часу оновлення	datetime		
contacts	id	Ідентифікатор	integer	РК	Не пустий
	name	Ім'я	varchar		Не пустий
	email	Пошта	varchar		Не пустий
	phone	Телефон	varchar		Не пустий
	subject	Тема	varchar		
	message	Повідомлення	text		Не пустий
	created_at	Відмітка часу створення	datetime		
	updated_at	Відмітка часу оновлення	datetime		
courses	id	Ідентифікатор	integer	РК	Не пустий
	title	Заголовок	varchar		Не пустий
	author	Автор	varchar		Не пустий

	date	Дата	date		Не пустий
	description	Опис	text		Не пустий
	image	Зображення курсу	varchar		
	created_at	Відмітка часу створення	datetime		
	updated_at	Відмітка часу оновлення	datetime		
course_videos	id	Ідентифікатор	integer	PK	Не пустий
	course_id	Ідентифікатор курсу	integer	FK	Не пустий
	url	Посилання на відео	varchar		Не пустий
	title	Заголовок	varchar		Не пустий
	thumbnail	Заставка	varchar		Не пустий
	created_at	Відмітка часу створення	datetime		
	updated_at	Відмітка часу оновлення	datetime		
	id	Ідентифікатор	integer	PK	Не пустий
	user_id	Ідентифікатор користувача	integer	FK	Не пустий

user_favourite_courses	course_id	Ідентифікатор курсу	integer	FK	Не пустий
	created_at	Відмітка часу створення	datetime		
	updated_at	Відмітка часу оновлення	datetime		
personal_access_token	id	Ідентифікатор	integer	PK	Не пустий
	tokenayble_type	Тип сутності, що підлягає токенизації	varchar		Не пустий
	tokenayble_id		varchar		Не пустий
	name	Ім'я	varchar		Не пустий
	token	Токен	varchar		Не пустий
	abilities	Можливості, пов'язані з токеном	text		
	last_used_at	Токен збереження сесії	datetime		
	expires_at	Відмітка часу терміну дії	datetime		
	created_at	Відмітка часу створення	datetime		

	updated_at	Відмітка часу оновлення	datetime		
password_reset_token	email	Ідентифікатор	varchar	PK	Не пустий
	token	Ідентифікатор користувача	varchar		Не пустий
	created_at	Відмітка часу створення	datetime		
failed_jobs	id	Ідентифікатор	integer	PK	Не пустий
	uuid	Універсальний унікальний ідентифікатор	varchar		Не пустий Унікальний
	connection	Назва з'єднання	text		Не пустий
	queue	Назва черги	text		Не пустий
	payload	Серійне представлення даних	text		Не пустий
	exception	Виняток, що виник під час виконання завдання.	text		Не пустий
	failed_at	Мітка часу невдалого виконання завдання.	datetime		Не пустий

1.3.9 Розробка клієнтської частини

Розробка клієнтської частини розпочинається зі створення компонентів майбутнього веб-сайту.

Однофайлові компоненти (також відомі як файли *.vue, скорочено SFC) - це спеціальний формат файлів, який дозволяє інкапсулювати шаблон, логіку та стилізацію компонента Vue в одному файлі.

```
<script>
export default {
  data() {
    return {
      greeting: 'Hello World!'
    }
  }
}
</script>

<template>
<p class="greeting">{{ greeting }}</p>
</template>

<style>
.greeting {
  color: red;
  font-weight: bold;
}
</style>
```

Рисунок 1.35 Приклад однофайлового компонента

Блоки <template>, <script> і <style> інкапсулюють і розміщують вигляд, логіку і стилізацію компонента в одному файлі.

Проект містить такі однофайлові компоненти, як верхня панель, бічна панель, нижня панель та інші важливі компоненти сайту (див. рис. 1.36).

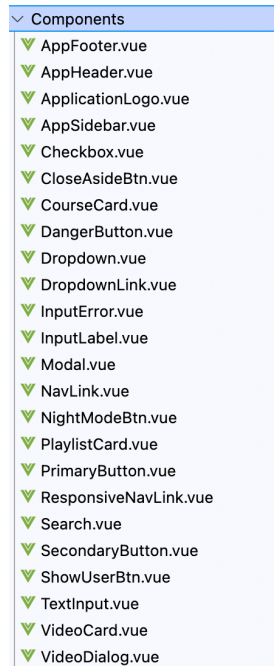


Рисунок 1.36 Компоненти проекту

Vuex — шаблон керування станом для Vue.js. Центром кожного додатку Vuex є "store" - сховище. "Сховище" - це, по суті, контейнер, в якому зберігається стан додатку.

Єдиний спосіб фактично змінити стан у сховищі Vuex – це здійснити мутацію. Мутації Vuex дуже схожі на події: кожна мутація має тип рядка та обробник. Функція обробника – це місце, де ми виконуємо фактичні зміни стану, і вона отримує стан як перший аргумент:

```
const store = createStore({
  state: {
    isNightModeActive: true,
    isShowAside: false,
    isProfileVisible: false,
    user: null,
    email: "",
    isShowVideoModal: false,
    modalVideo: {
      id: null,
      title: "",
      description: "",
      url: "",
    }
  },
  mutations: {
    toggleNightMode(state) {
      state.isNightModeActive = !state.isNightModeActive;
    },
  },
});
```

Рисунок 1.37 Приклад мутації у Vuex

Сторінки представляють собою різні вікна або потоки додатку. Кожна сторінка зазвичай складається з комбінації компонентів і відповідає за відображення загального вигляду і вмісту відповідного вікна. Сторінки використовуються для створення основної структури та навігаційного простору додатку.

Майбутній сайт міститиме сторінки представлені на малюнку 1.38.

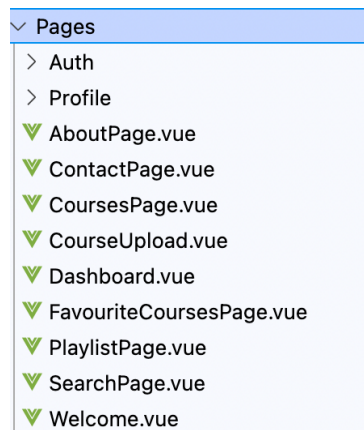


Рисунок 1.38 Сторінки проекту

Директорія Auth містить сторінки реєстрації, логіну, відновлення паролю та підтвердження електронної пошти, забезпечуючи необхідні функціональні можливості для автентифікації користувача та управління обліковим записом.

Директорія Profile включає сторінку персонального облікового запису користувача, де він може переглядати та оновлювати свої дані. Додатково тут знаходяться компоненти для зміни пароля та модифікації особистої інформації, що дає користувачам змогу керувати своїм обліковим записом.

AboutPage призначена для надання детальної інформації про компанію. Користувачі можуть дізнатися про компанію, її історію, місію, цінності та будь-які інші важливі деталі, які допоможуть їм краще зрозуміти організацію.

ContactPage слугує для користувачів засобом зв'язку з адміністрацією або службою підтримки. Вона містить контактну форму та контактну інформацію, зокрема адресу електронної пошти та номер телефону.

CoursesPage слугує своєрідним центром, який демонструє повний перелік доступних курсів. Користувачі можуть ознайомитися з різноманітним переліком пропонуванних курсів, а також скористатися фільтрами чи опціями сортування, щоб знайти курси, які відповідають їхнім інтересам чи потребам.

CourseUpload - це спеціалізована сторінка, доступна лише адміністратору. Вона дозволяє завантажувати нові курси на платформу, забезпечуючи постійне розширення каталогу курсів і розширюючи можливості навчання для користувачів.

Dashboard - це перехідна сторінка, на яку користувачі потрапляють після реєстрації або входу в систему. Вона забезпечує персоналізований і детальний огляд облікового запису користувача, відображаючи відповідну інформацію, сповіщення та ярлики для доступу до часто використовуваних функцій.

Favourites - це спеціальна сторінка для доступу до збережених або доданих до закладок курсів, а також для керування ними. Завдяки цьому можна легко відстежувати курси, які вас цікавлять, і мати до них швидкий доступ для подальшого перегляду.

PlaylistPage дозволяє користувачам переглядати детальні описи курсів та відеоуроки в межах певного курсу. Це забезпечує структурований і захоплюючий процес навчання, полегшуючи безперешкодну навігацію контентом курсу.

SearchPage розроблена як комплексний інструмент пошуку курсів, що дозволяє користувачам шукати курси за ключовими словами, темами або конкретними критеріями. Вона допомагає користувачам швидко знайти відповідні курси, які відповідають їхнім уподобанням або навчальним цілям.

WelcomePage слугує початковою цільовою сторінкою для користувачів. Вона пропонує теплий і цікавий вступ до платформи, надає необхідну інформацію, заохочує користувачів до подальших досліджень і скеровує їх до наступних кроків у їхній навчальній подорожі.

1.3.10 Розробка серверної частини

Реалізація бази даних

Перш за все, необхідно спроектувати структуру бази даних, створивши файл міграції в Laravel. Для створення міграції бази даних слід скористатися командою “php artisan make:migration create_coures_table”. Клас міграції містить два методи: up і down. Метод up використовується для додавання нових таблиць, стовпців або індексів до бази даних, в той час як метод down скасовує операції, виконані методом up [12].

```
return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('courses', function (Blueprint $table) {
            $table->id();
            $table->string('title');
            $table->string('author');
            $table->date('date');
            $table->text('description');
            $table->string('image')->nullable();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('courses');
    }
};
```

Рисунок 1.39 Міграція для створення таблиці Courses

Щоб виконати всі наявні міграції треба скористатися командою “php artisan migrate”.

Створення моделей та визначення зв'язків.

Наступним кроком в процесі реалізації бази даних є створення моделей та визначення зв'язків.

Моделлю в Laravel називають таблицю бази даних, яка керує взаємодією з базою даних. Згенерувати модель для таблиці "users" можна за допомогою команди "php artisan make:model Course". Згенерована модель буде розташована в каталозі "app" і може бути відредагована відповідно до потреб.

```

class Course extends Model
{
    use HasFactory;

    public $fillable = [
        'title', 'author', 'date', 'description', 'image'
    ];

    public function videos()
    {
        return $this->hasMany(CourseVideo::class);
    }
}

```

Рисунок 1.40 Модель Course

Клас "Course" охоплює клас "Model", який є базовим класом для моделей Laravel, що забезпечує основну функціональність.

Властивість \$fillable визначає, які поля слід враховувати, коли користувач вводитиме або оновлюватиме дані. У цьому випадку вона включає такі поля, як "title", "author", "date", "description" та "image".

Метод "videos" визначає зв'язок "один-до-багатьох" між моделлю "Course" і моделлю "CourseVideo". Для встановлення зв'язку використовується метод hasMany, який вказує на те, що курс може мати декілька пов'язаних відео.

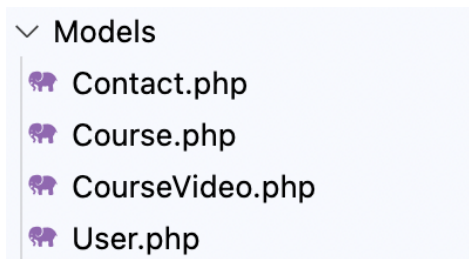


Рисунок 1.41 Перелік моделей проекту

Створення контролера

Замість того, щоб визначати всю логіку обробки запитів як замикання у файлах маршрутів, можна організувати цю поведінку за допомогою класів "контролерів". Контролери можуть групувати пов'язану логіку обробки запитів в одному класі. Щоб швидко згенерувати новий контролер, необхідно скористатися

командою “php artisan make:controller CourseController” [13]. За замовчуванням контролери зберігаються в каталозі app/Http/Controllers.

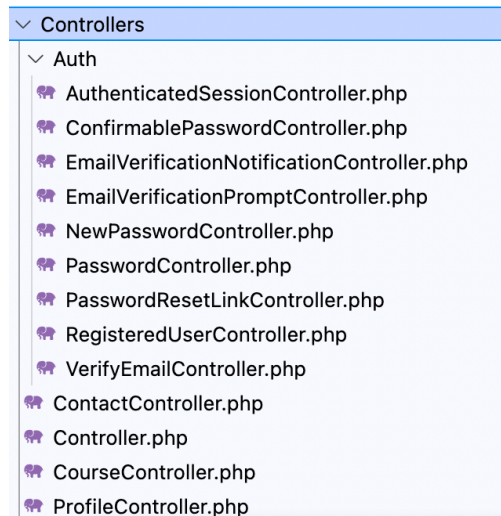


Рисунок 1.42 Перелік контролерів проекту

```
public function store(Request $request): \Illuminate\Http\RedirectResponse
{
    $request->validate([
        'title' => 'required',
        'author' => 'required',
        'date' => 'required|date',
        'description' => 'required',
        'image' => 'file|mimes:jpg,png,jpeg,gif,svg|max:2048',
    ]);

    $course = new Course();
    $course->title = $request->input('title');
    $course->author = $request->input('author');
    $course->date = $request->input('date');
    $course->description = $request->input('description');

    // Handle course image upload
    if ($request->hasFile('image') && $request->file('image')->isValid()) {
        $filePath = $request->file('image')->store('uploads/courses', 'public');
        $course->image = '/storage/' . $filePath;
    }

    $course->save();

    return Redirect::back()->with([
        'data' => 'Course uploaded.',
    ]);
}
```

Рисунок 1.43 Фрагмент коду CourseController

Вищенаведений фрагмент коду демонструє метод "store" в межах "CourseController".

Метод "store" обробляє валідацію та зберігання нового курсу. Він перевіряє дані вхідного запиту, створює новий екземпляр "Course", привласнює вхідні значення, обробляє завантаження зображень, якщо це необхідно, зберігає курс в базі даних і перенаправляє назад з флеш-повідомленням про успішне завантаження курсу.

Визначення маршрутів

Laravel використовує досить простий і зрозумілий метод побудови маршрутів, найпростіші маршрути складаються з URI і функції замикання (closure). Всі маршрути Laravel визначені у файлах маршрутів, які знаходяться у каталозі routes. Файл routes/web.php визначає маршрути для веб інтерфейсу.

```
Route::get('/courses', [CourseController::class, 'showCoursesPage'])->name('courses');
```

Рисунок 1.44 Приклад маршруту у файлі web.php

Представлений вище маршрут визначає GET-запит на шлях '/courses'. Коли виконується GET-запит, Laravel викликає метод showCoursesPage в CourseController, який відповідає за відображення сторінки з курсами.

Посередники

Посередники надають зручний механізм для перевірки та фільтрації HTTP-запитів, що надходять до додатку. Наприклад, Laravel містить посередник, який перевіряє, чи авторизований користувач вашого додатку. Якщо користувач не автентифікований, посередник перенаправляє його на екран входу до вашого додатку. Однак, якщо користувач автентифікований, посередник дозволить запиту продовжити роботу з сайтом [14].

Щоб створити посередник, треба скористатися командою "php artisan make:middleware Name". Усі посередники розміщені в каталозі app/Http/Middleware.

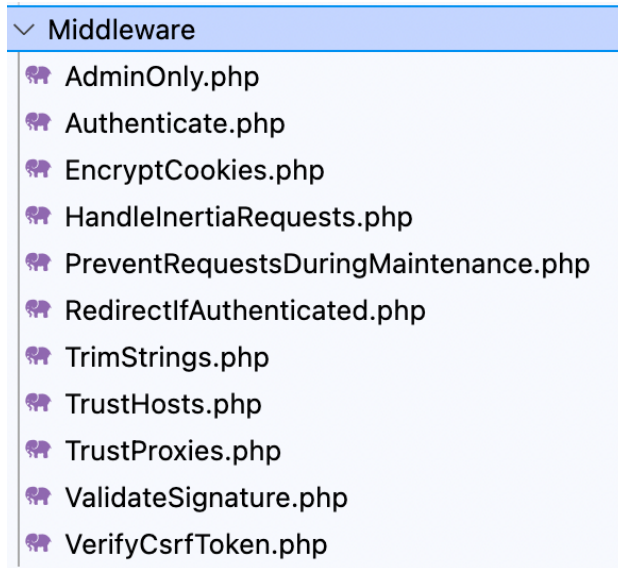


Рисунок 1.45 Перелік посередників проекту

Щоб зареєструвати посередник у псевдонімах посередників, треба перейти до каталогу `app/Http/` і відкрити файл `Kernel.php`. Далі слід додати посередник разом з псевдонімом та шляхом до нього у якості елемента масиву `$middlewareAliases`.

```
protected $middlewareAliases = [
    'auth' => \App\Http\Middleware\Authenticate::class,
    'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
    'auth.session' => \Illuminate\Session\Middleware\AuthenticateSession::class,
    'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
    'can' => \Illuminate\Auth\Middleware\Authorize::class,
    'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
    'password.confirm' => \Illuminate\Auth\Middleware\RequirePassword::class,
    'signed' => \App\Http\Middleware\ValidateSignature::class,
    'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
    'verified' => \Illuminate\Auth\Middleware\EnsureEmailsVerified::class,
];
```

Рисунок 1.46 Масив `$middlewareAliases`

1.3.11 Тестування розробленого сайту

Тестування - це детальна перевірка веб-сайту або додатку на предмет наявності потенційних недоліків. Іншими словами, це повне тестування веб-сайтів перед їхнім впровадженням. Під час тестування оцінюється ступінь зручності та

зовнішньої привабливості сервісу для користувача, визначається швидкість доступу до необхідної інформації та надійність використання.

1) Тестування функціональності

- Перевірка відсутності непрацюючих сторінок або недійсних посилань.
- Ввод невірних даних для виконання негативного тестування.
- Перевірка робочого циклу системи.
- Перевірка цілісності даних.

При перевірці не було виявлено непрацюючих сторінок або недійсних посилань. При введенні невірних даних форми реагували належним чином показуючи помилку (див. рис. 1.47).

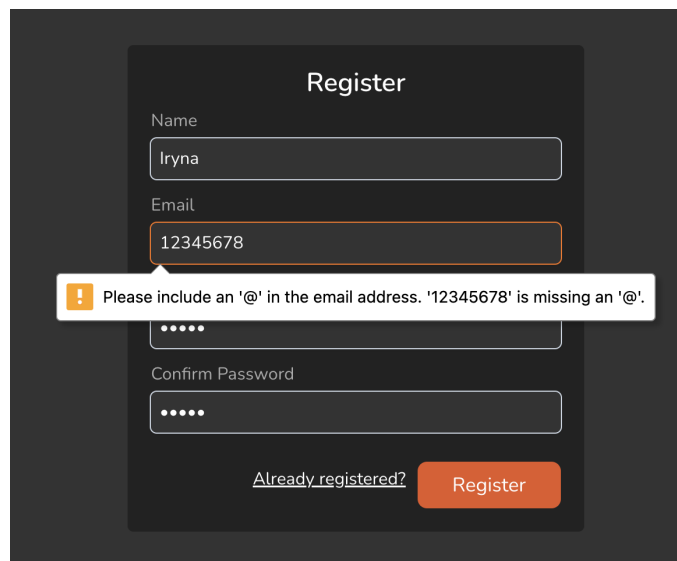


Рисунок 1.47 Негативне тестування форми реєстрації

2) Юзабіліті-тестування - перевірка зручності користування додатком.

Юзабіліті-тест показав, що сайт зручний у використанні, має просту та інтуїтивно зрозумілу навігацію, приємну кольорову гаму та зрозумілу логічну структуру.

3) Тестування на сумісність

- Сумісність з браузерами
- Сумісність з операційною системою
- Сумісність з різними пристроями, такими як ноутбук, мобільний телефон тощо.

Тестування було виконано у двох операційних системах - Windows та MacOS, трьох браузерів - Safari, Chrome та Internet Explorer. Перевірка на сумісність була перевірена на таких пристроях як ноутбук, планшет та мобільний телефон. При тестуванні виявлено, що веб-сайт однаково відображається у всіх браузерах, операційних системах та на всіх пристроях.

					<i>КГ 06. 21 000. 00 ДП ПЗ</i>	59
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ЕКОНОМІЧНА ЧАСТИНА

В рамках програмної реалізації кваліфікаційної роботи було поставлено завдання розробити адаптивний веб-сайт онлайн навчання.

Онлайн-навчання стає дедалі актуальнішим, оскільки все більше людей віддають перевагу навчанню у власному темпі та у зручний для них час. У результаті виникла потреба в добре розроблених платформах онлайн-навчання, які можуть запропонувати високоякісні освітні матеріали, щоб задовольнити зростаючий попит на онлайн-освіту. Крім того, розробка відкритої навчальної платформи розширить параметри навчання в онлайн-середовищі, дозволивши різноманітним користувачам брати участь у навчальному процесі, як початківцям так і досвідченим розробникам.

Оскільки створюваний сайт не має комерційної спрямованості, у процесі оцінки його ефективності будуть визначені наступні види ефективностей: функціональна та соціальна.

Крім цього, в процесі роботи будуть визначені загальні витрати на створення сайту, що включають в себе витрати на розробку сайту та витрати на впровадження сайту.

Таким чином, оцінка ефективності створюваного некомерційного сайту враховує функціональну та соціальну ефективність, а також аналіз загальних витрат на його створення та впровадження. Це дозволяє забезпечити оптимальну якість роботи сайту та досягнення позитивного впливу на його користувачів та спільноту.

Загальні витрати (B_3) на створення сайту складаються з декількох параметрів:

$$B_3 = B_p + B_v ,$$

де B_p – витрати на розробку сайту;

B_v – витрати на впровадження сайту;

					<i>КГ 06. 21 000. 00 ДП ПЗ</i>	60
Зм.	Арк.	№ докум.	Підпис	Дата		

Витрати на розробку сайту (V_p) є одноразовими та складаються з вартості наступних видів робіт зі створення сайту :

1. Розробка дизайну сайту: розробка макетів дизайну для всіх сторінок сайту; підбір кольорової палітри; розробка логотипу;
2. Розробка клієнтської частини сайту: створення компонентів та сторінок; реалізація адаптивності веб-сторінок;
3. Розробка серверної частини сайту: реалізація бази даних; розробка модулів аутентифікації; побудова маршрутів; розробка модулів пошуку, каталогу курсів, форми зв'язку;
4. Тестування веб-сайту.

Для визначення витрат на розробку сайту (V_p) розраховуємо оплату праці виконавців, безпосередньо притягнених до її виконання. Для реалізації проекту Web-системи використовувалися наступні спеціалісти: дизайнер та розробник.

Для визначення трудомісткості розробки сайту (V_p) складаємо план-графік по розробці web-сайту і тривалості виконання робіт. Розподіл робіт по етапах і видах виконавців наведено в таблиці 3.1.

Таблиця 3.1 – План-графік по розробці Web-сайту

№	Назва етапу	Час виконання (годин)	Посада виконавця
1	Розробка дизайну сайту	25	дизайнер
2	Розробка клієнтської частини сайту	40	розробник
3	Розробка серверної частини сайту	45	розробник
4	Тестування веб-сайту	10	розробник
ВСЬОГО:		120 годин	

Розрахунок трудомісткості здійснюємо в наступній послідовності:

1. Складаємо перелік всіх етапів і видів робіт, які необхідно виконати в ході даної розробки. Після узгодження з керівником проекту допускаємо виключення, доповнення, об'єднання окремих етапів і видів робіт;

2. По кожному виду робіт визначаємо кваліфікаційний рівень виконавців. В разі виконання однієї роботи виконавцями різної кваліфікації, робота розподіляється на ряд паралельних конкретних робіт для кожної категорії виконавця.

В умовах відсутності нормативної бази тривалість виконання окремих робіт розраховуємо на основі вірогідних оцінок робіт, що задаються виконавцями.

Розмір заробітної плати розраховуємо виходячи з чисельності різних категорій виконавців, трудомісткості, що витрачається ними на виконання різних видів робіт, а також їх середньої заробітної плати (ставки) за годину.

При визначенні вартості виконуваних робіт орієнтуємося на ціни, представлені на сайтах фірм, що спеціалізуються в сфері створення та модернізації web-ресурсів. Витрати на заробітну плату приведені в таблиці 3.2.

Таблиця 3.2 – Витрати на заробітну плату

№	Персонал	Етапи розробки	Кількість робочих годин	Погодинна ставка, грн.	Заробітна плата, грн.
1	Дизайнер	Розробка дизайну сайту	25	60	1500
2	Розробник	Розробка клієнтської частини сайту; Розробка серверної частини сайту; Тестування веб-сайту;	95	75	7125

ВСЬОГО:	$V_{зп} = 8625$ грн.
---------	----------------------

До складу витрат на оплату праці також включаються податки, збори і інші обов'язкові платежі, встановлені системою оподаткування що діє. Розмір єдиного соціального внеску складає 22% від заробітної плати, розраховується за наступною формулою:

$$V_{есв} = V_{зп} \times 0,22 = 8625 \times 0,22 = 1897,5 \text{ грн.}$$

Загальні витрати (V_p) на розробку веб-сайту розраховуються як сума витрат на заробітну плату праці персоналу ($V_{зп}$) та єдиного соціального внеску ($V_{есв}$):

$$V_p = V_{зп} + V_{есв} = 8625 + 1897,5 = 10522,5 \text{ грн.}$$

Витрати на впровадження сайту (V_b) складаються з двох складових:

- витрати на реєстрацію доменного імені на 1 рік (V_{b1});
- витрати на реєстрацію в пошукових системах (V_{b2}), наприклад, Yandex, Google, Rambler и т.п.)

$$V_b = V_{b1} + V_{b2} = 486 + 1240 = 1726 \text{ грн.}$$

Загальні витрати (V_z) на розробку та впровадження веб-сайту розраховуються за наступною формулою:

$$V_z = V_p + V_b = 10522,5 + 1726 = 12248,5 \text{ грн.}$$

Функціональна ефективність проявляється:

1. в забезпеченні повноти та точності інформації про курси, та інші деталі, які можуть бути корисними для користувачів;
2. в доступності навчального матеріалу для користувачів у будь-який час і в будь-якому зручному для них місці;
3. у використанні оптимізованих технологічних процесів для забезпечення швидкості та надійності роботи сайту;
4. в забезпеченні зручного та простого у використанні інтерфейсу для користувачів;

5. у готовності сайту до масштабування для забезпечення росту числа користувачів і обробки великої кількості даних.

Соціальна ефективність, або соціальний ефект у свою чергу, оцінюється на основі досягнення соціальних цілей та позитивного впливу на користувачів і спільноту та виражається в наступному:

1. у забезпеченні доступності освіти для широкого кола людей незалежно від їх місця проживання, фінансового стану чи фізичних обмежень. Це може бути особливо корисно для людей, які не мають можливості відвідувати традиційні навчальні заклади, наприклад, через зайнятість, сімейні обставини або обмежений фізичний доступ;
2. у наданні користувачам можливості самостійно планувати свій навчальний графік і вивчати матеріал у власному темпі.

					<i>КГ 06. 21 000. 00 ДП ПЗ</i>	
Зм.	Арк.	№ докум.	Підпис	Дата		64

3 ОХОРОНА ПРАЦІ

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних та лікувально - профілактичних заходів, спрямованих на збереження здоров'я та працездатності людини.

Метою охорони праці є забезпечення здорових та безпечних умов праці.

Завдання охорони праці полягає в тому, щоб звести до мінімуму ймовірність травматизму та виникнення професійних захворювань.

3.1 Аналіз небезпечних і шкідливих факторів, що впливають на працівника при розробці програмного комплексу

Небезпечні та шкідливі виробничі фактори залежно від природи виникнення поділяють на чотири групи: фізичні, хімічні, психофізіологічні, біологічні.

При роботі в приміщенні на програміста можуть негативно діяти такі

- підвищена і знижена температура повітря;
- підвищена і знижена вологість повітря;
- недостатня освітленість робочого місця;
- шум, що перевищує допустимі норми;
- підвищений рівень статичної електрики;
- небезпека ураження електричним струмом;

До хімічно небезпечних факторів, що діють на програміста під час роботи відносяться наступні:

- виникнення активних частинок внаслідок іонізації повітря під час роботи комп'ютера.

До психологічно шкідливих чинників, що впливають на програміста протягом роботи, належать наступні:

- нервово-емоційні перевантаження;
- розумова напруга;

					<i>КГ 06. 21 000. 00 ДП ПЗ</i>	65
Зм.	Арк.	№ докум.	Підпис	Дата		

- перенапряга зорового аналізатора.

3.2 Гігієнічні вимоги до виробничого середовища

3.2.1 Вимоги до приміщення

Площа приміщення на одне робоче місце має становити щонайменше 6,0 м², об'єм – 20.0 куб.м. Робоче місце повинно знаходитися на відстані щонайменше 1 м від стіни з вікном і не менше 1,4 м від звичайної стіни, відстань між боковими стінками комп'ютерів має становити щонайменше 1,2 м, а відстань між тильною стінкою комп'ютера та екраном іншого комп'ютера має бути не менше 2,5 м. Слід також подбати про те, щоб кольори приміщення були гармонійними. Найбільш сприятливими для нервової системи є бліді пастельні кольори. Яскраві та контрастні поєднання можуть викликати втому та роздратування.

3.2.2 Вимоги до організації робочого місця

Робочі місця повинні бути розташовані так, щоб світло падало з лівого боку, екран монітора повинен знаходитися на оптимальній відстані від очей користувача (60-70 см), екран повинен бути розташований під кутом +30° до нормальної лінії погляду працівника для комфортного спостереження за вертикальною площиною, а поверхня клавіатури повинна мати антистатичні властивості.

3.2.3 Вимоги до освітлення приміщення

Природне освітлення повинно забезпечуватися переважно через бічні світлові прорізи, орієнтовані на північ або північний схід. Коефіцієнт природного освітлення повинен становити не менше 1,5%. Віконні прорізи повинні бути забезпечені регульованими пристроями для відкривання, а також шторами або зовнішніми козирками.

					<i>КГ 06. 21 000. 00 ДП ПЗ</i>	66
Зм.	Арк.	№ докум.	Підпис	Дата		

Штучне освітлення повинно забезпечуватися загальною рівномірною системою освітлення, що складається з ряду світильників безперервної або періодичної дії, розташованих збоку (переважно з лівого боку) від робочого місця паралельно лінії зору користувача комп'ютера.

Рівень освітленості на робочому столі користувача в зоні розміщення документів повинен становити 300-500 лк. Коли цей рівень освітленості неможливо забезпечити системою загального освітлення, то дозволяється використання світильників місцевого освітлення, однак при цьому не повинно бути відблисків на поверхні екрана (яскравість відблисків не має перевищувати 40 кд/м²) і перевищення рівня його освітленості більш ніж 300 лк.

3.2.4 Мікроклімат у приміщенні

Саме температура і вологість повітря мають безпосередній вплив на загальне самопочуття, стан слизових оболонок очей, верхніх дихальних шляхів і шкіри людини. Низька вологість збільшує тривалість "зависання" пилу в повітрі приміщення.

У теплий період року температура повітря має бути у межах 22-25 °С, швидкість руху повітря — до 0,1 м/с, відносна вологість повітря — 40-60%.

У холодний період року температура повітря може коливатися у межах 21-24 °С, швидкість руху повітря — до 0,1 м с, вологість повітря — 40-60%.

Задля забезпечення та автоматичного підтримання в приміщенні незалежно від зовнішніх умов оптимальних значень температури, вологості, чистоти та швидкості руху повітря, в холодну пору року використовується водяне опалення, в теплу пору року застосовується кондиціонування повітря.

Джерелом хімічних речовин у повітрі робочого простору можуть бути матеріали, що використовувалися для ремонту та оздоблення приміщення. Найбільш поширеною речовиною, яка може виділятися з полімерних матеріалів, є формальдегід. Вміст формальдегіду в повітрі робочої зони повинен становити 0,5 ГДК.

3.2.5 Шум та вібрації

Внаслідок тривалого впливу шуму на організм людини погіршуються умови праці. У людей, які працюють в умовах постійного впливу шуму, з'являється роздратованість, головний біль, запаморочення, погіршення пам'яті тощо. Норма допустимого рівня шуму на робочому місці програміста не повинна перевищувати 50 дБ.

Щоб знизити рівень шуму, стіни і стелі приміщень, де встановлені комп'ютери, можна облицювати звукопоглинальними матеріалами. Рівень вібрації в приміщеннях комп'ютерних центрів можна знизити, встановивши устаткування на спеціальні віброізолятори.

3.2.6 Електробезпека

Основні правила електробезпеки, яких слід дотримуватися під час роботи з комп'ютером, включають наступне:

1. Користуйтеся тільки перевіреними та надійними джерелами живлення.
2. Уникайте перевантаження електричних мереж і розеток.
3. Регулярно перевіряйте дроти живлення та роз'єми на предмет пошкоджень, тріщин або видимих ознак зносу.
4. Переконайтеся, що комп'ютер і його компоненти зберігаються в сухому і захищеному від вологи місці. Уникайте проливання рідини на апаратне забезпечення комп'ютера.
5. Без необхідних знань і досвіду не розбирайте апаратне забезпечення комп'ютера або периферійні пристрої.
6. Перед чищенням або підключенням периферійних пристроїв завжди вимикайте комп'ютер і від'єднуйте його від джерела живлення.

3.3 Пожежна безпека

При використанні комп'ютера важливо дотримуватися наступних правил пожежної безпеки:

- Не можна залишати комп'ютер увімкненим без нагляду. Обов'язково вимикайте його, коли залишаєте робоче місце.
- Регулярно очищати комп'ютер від пилу та бруду. Пил може стати горючим матеріалом і призвести до перегріву компонентів.
- В приміщенні, де знаходиться комп'ютер, встановити і підтримувати в робочому стані пожежну сигналізацію та вогнегасники.
- Регулярно перевіряти стан електропроводки та розеток. При виявленні будь-яких несправностей звернутися до кваліфікованого електрика для усунення несправностей.
- Уникати надмірного нагрівання компонентів комп'ютера, забезпечуючи належну вентиляцію. Не закривати вентиляційні отвори в корпусі комп'ютера і не ставити комп'ютер на м'які або світловідбиваючі поверхні.

У разі виникнення пожежі негайно вимкніть обладнання, відключіть електроживлення, повідомте про пожежу всіх співробітників і розпочинайте гасіння пожежі за допомогою наявних засобів пожежогашіння.

До засобів гасіння пожежі відносяться внутрішні пожежні водопроводи (крани –ПК), вогнегасники (вуглекислотні та порошкові), сухий пісок тощо.

Виробничі приміщення мають запасні виходи. Двері повинні мати освітлений надпис « Запасний вихід ». План евакуації вивішується на видному місці у основного виходу із приміщення.

У випадку виникнення пожежі необхідно викликати по телефону 101 пожежну команду, евакуювати людей із приміщення відповідно до плану евакуації і приступити до ліквідації пожежі.

ВИСНОВКИ

Впровадження передових технологій та інструментів розробки веб-додатків в процес розробки та впровадження адаптивного веб-сайту для онлайн-навчання відповідає зростаючому попиту на цю форму освіти. Швидкий прогрес технологій та можливість навчання власним темпом та в зручний для користувачів час стали важливими факторами популярності онлайн-навчання.

Розроблений адаптивний веб-сайт надає користувачам можливість отримувати доступ до навчальних курсів з будь-якого пристрою, забезпечуючи при цьому збереження функціональності та зручності використання. Відкрита навчальна платформа розширює можливості онлайн-навчання, надаючи змогу участі різним категоріям користувачів, від новачків до досвідчених розробників.

Процес розробки веб-сайту включав аналіз предметної області, вивчення існуючих аналогічних веб-додатків та використання передових засобів розробки, таких як Vue.js та Laravel. Результатом цієї роботи є готова програмна реалізація браузерної платформи для онлайн-навчання, яка використовує передові технології та інструменти для розробки веб-додатків.

Використання HTML/CSS/JS, фреймворку Vue.js, бібліотеки Vuex та InertiaJs, серверного фреймворку Laravel, CSS-фреймворку Tailwind CSS, редактора коду Microsoft Visual Studio Code, графічного редактора Figma та системи контролю версій Git сприяло успішному втіленню цього проекту та дозволило досягти поставленої мети.

					<i>КГ 06. 21 000. 00 ДП ПЗ</i>	70
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ресурс Інтернету: ITVDN. URL: <https://itvdn.com> (дата звернення 02.05.2023).
2. Ресурс Інтернету: Webner Solutions Blog. "Visual Studio Code: An Overview and Key Features." URL: <https://blog.webnersolutions.com/visual-studio-code/> (дата звернення 04.05.2023).
3. Ресурс Інтернету: Webuild.io. "Why Use Figma for Digital Product Design?" URL: <https://webuild.io/why-use-figma-for-digital-product-design> (дата звернення 04.05.2023).
4. Ресурс Інтернету: General Assembly Blog. "What Is a JavaScript Framework?" URL: <https://generalassemb.ly/blog/what-is-a-javascript-framework/> (дата звернення 04.05.2023).
5. Ресурс Інтернету: Monocubed Blog. "Advantages of Vue.js." URL: <https://www.monocubed.com/blog/advantages-of-vue-js/> (дата звернення 04.05.2023).
6. Ресурс Інтернету: TutorialsPoint. "Laravel - Overview." URL: https://www.tutorialspoint.com/laravel/laravel_overview.htm (дата звернення 04.05.2023).
7. Ресурс Інтернету: HubSpot Blog. "What Is Tailwind CSS?" URL: <https://blog.hubspot.com/website/what-is-tailwind-css> (дата звернення 04.05.2023).
8. Ресурс Інтернету: JavaTpoint. "Git Tutorial." URL: <https://www.javatpoint.com/git> (дата звернення 04.05.2023).
9. Ресурс Інтернету: Atomic Design by Brad Frost. "Chapter 2: Atomic Design Methodology." URL: <https://atomicdesign.bradfrost.com/chapter-2> (дата звернення 05.05.2023).
10. Ресурс Інтернету: Laravel Documentation. "Laravel - Structure." URL: <https://laravel.com/docs/10.x/structure> (дата звернення 10.05.2023).

					<i>КГ 06. 21 000. 00 ДП ПЗ</i>	
Зм.	Арк.	№ докум.	Підпис	Дата		71

11. Ресурс Інтернету: GeeksforGeeks. "Introduction of ER Model in DBMS."
URL:<https://www.geeksforgeeks.org/introduction-of-er-model>
(дата звернення 12.05.2023).
12. Ресурс Інтернету: Laravel Documentation. "Laravel - Migrations." URL:
<https://laravel.com/docs/10.x/migrations> (дата звернення 19.05.2023).
13. Ресурс Інтернету: Laravel Documentation. "Laravel - Controllers." URL:
<https://laravel.com/docs/10.x/controllers> (дата звернення 20.05.2023).
14. Ресурс Інтернету: Laravel Documentation. "Laravel - Middleware." URL:
<https://laravel.com/docs/10.x/middleware> (дата звернення 21.05.2023).

					<i>КГ 06. 21 000. 00 ДП ПЗ</i>	
Зм.	Арк.	№ докум.	Підпис	Дата		72

Розробка та впровадження адаптивного веб-сайту онлайн навчання

Виконала: студентка групи 4КГ-06
Мурзак І.В.
Керівник: Кунуп Т.В.

Мета дослідження. Актуальність

Метою дипломної роботи є отримання готової програмної реалізації браузерної платформи онлайн навчання, з використанням сучасних технологій та інструментів розробки веб-додатків.

Об'єктом дослідження є проектування, розробка та впровадження адаптивного веб-сайту онлайн навчання.

Предметом дослідження є засоби та технології створення адаптивного веб-сайту онлайн навчання.

Основні завдання дипломної роботи

1. Провести аналіз предметної області.
2. Розглянути та проаналізувати існуючі аналогічні веб-додатки.
3. Розглянути особливості сучасних засобів по розробці веб-додатків, виділити їх особливості, недоліки та переваги.
4. Розробити користувацький інтерфейс для веб - сайту.
5. Розробити серверну частину веб - сайту.
6. Виконати перевірку працездатності веб - сайту.
7. Впровадити веб-сайт у користування.

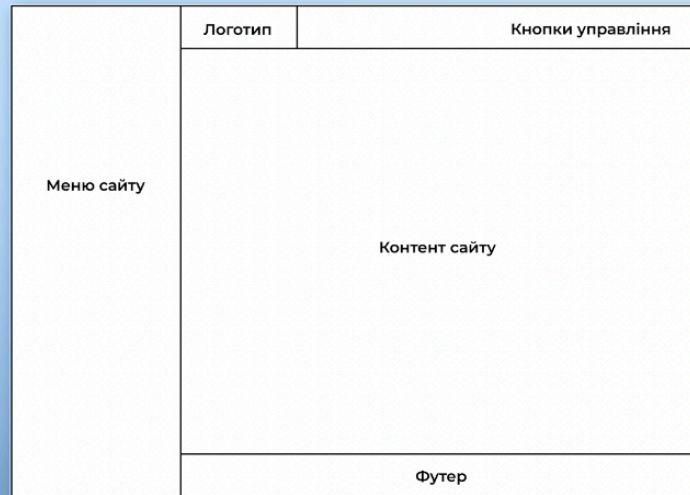
Аналіз існуючих програмних рішень



coursera

PROMETHEUS

Структура веб-сайту



Вибір програмного забезпечення

FIGMA

Графічний редактор для дизайну інтерфейсів

VS CODE

Редактор коду з розширеними можливостями

GIT

Система контролю версій



VUE JS

Створення інтерактивних користувацьких інтерфейсів

LARAVEL

Серверний фреймворк PHP

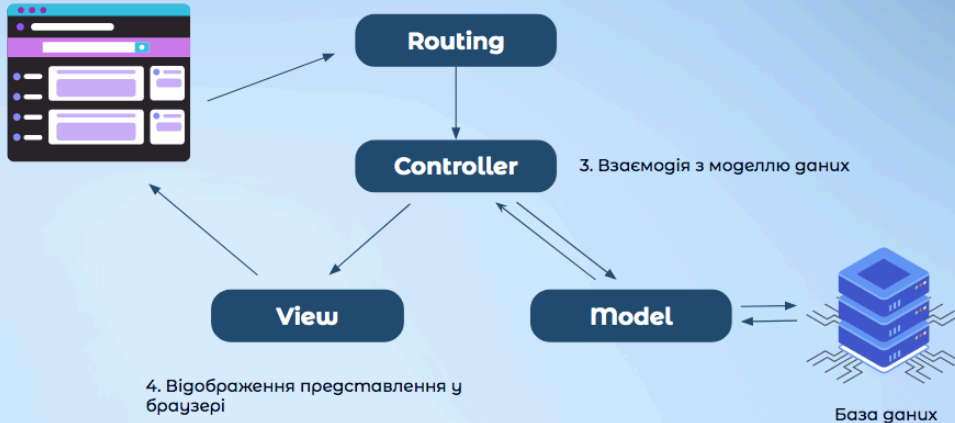
TAILWIND CSS

CSS-фреймворк для стилізації інтерфейсу

Архітектура MVC

1. Віправка запиту користувача

2. Маршрут до відповідного Laravel-контролера



Вимоги до веб-сайту. User Story

РОЛІ

- ГІСТЬ
- ЗАРЕЄСТРОВАННИЙ КОРИСТУВАЧ
- АДМІНІСТРАТОР

Як [роль користувача], я хочу [функціональність], щоб [переваги].

Як гість, я хочу мати змогу переглядати список доступних курсів, щоб визначити, чи варто мені реєструватися на сайті.

Як користувач, я хочу мати можливість змінити свій пароль, щоб підвищити безпеку свого облікового запису.

Як адміністратор, я хочу мати можливість змінювати інформацію про курс, щоб оновлювати та покращувати інформацію для користувачів.

Методологія атомарного дизайну



АТОМИ



МОЛЕКУЛИ



ОРГАНІЗМИ



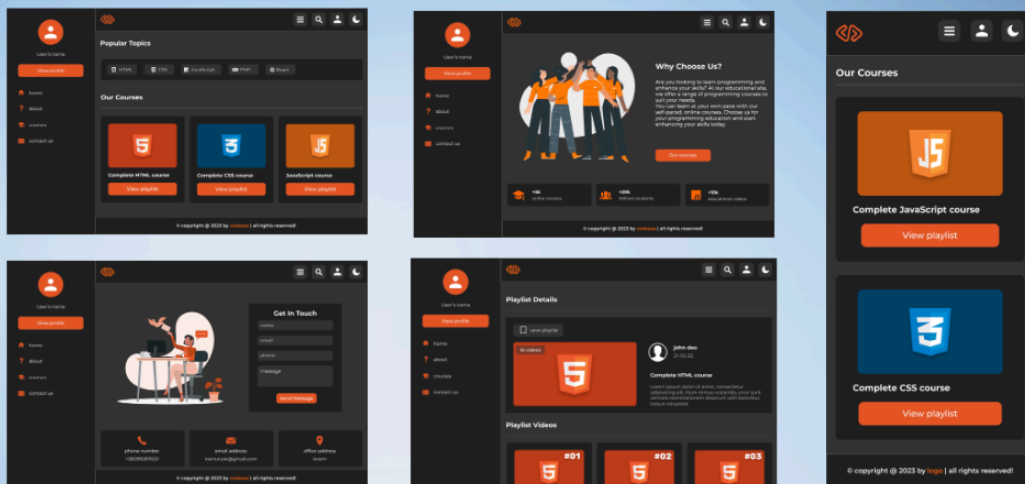
ШАБЛОНИ



СТОРІНКИ



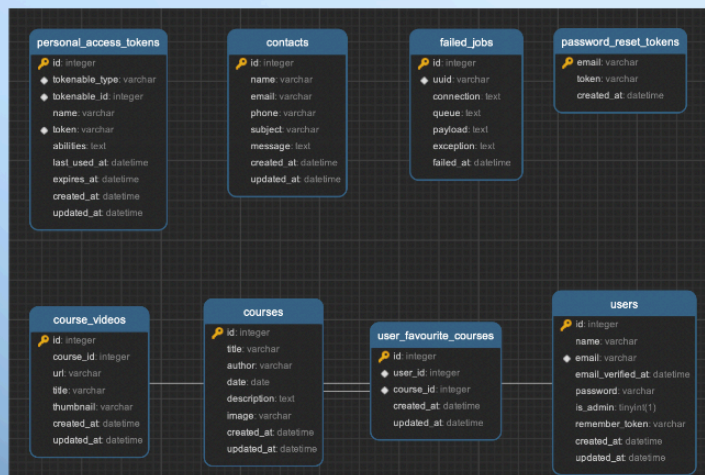
Розроблений користувацький інтерфейс



Вигляд веб-сайту на ноутбучі

На телефоні

ER-діаграма



Результати тестування веб-сайту

1) Тестування функціональності

При перевірці не було виявлено непрацюючих сторінок або недійсних посилань. При введенні невірних даних форми реагували належним чином показуючи помилку

2) Юзабіліті-тестування - перевірка зручності користування додатком.

Юзабіліті-тест показав, що сайт зручний у використанні, має просту та інтуїтивно зрозумілу навігацію, приємну кольорову гаму та зрозумілу логічну структуру.

3) Тестування на сумісність

Тестування було виконано у двох операційних системах - Windows та MacOS, трьох браузерах - Safari, Chrome та Internet Explorer. Перевірка на сумісність була перевірена на таких пристроях як ноутбук, планшет та мобільний телефон. При тестуванні виявлено, що веб-сайт однаково відображається у всіх браузерах, операційних системах та на всіх пристроях.

Висновки

Швидкий прогрес технологій та можливість навчання власним темпом та в зручний для користувачів час стали важливими факторами популярності онлайн-навчання.

Розроблений адаптивний веб-сайт надає користувачам можливість отримувати доступ до навчальних курсів з будь-якого пристрою, забезпечуючи при цьому збереження функціональності та зручності використання.

Використання HTML/CSS/JS, фреймворку Vue.js, бібліотеки Vuex та InertiaJs, серверного фреймворку Laravel, CSS-фреймворку Tailwind CSS, редактора коду Microsoft Visual Studio Code, графічного редактора Figma та системи контролю версій Git сприяло успішному втіленню цього проекту та дозволило досягти поставленої мети.

Дякую за увагу!

ВІДГУК

керівника на дипломний проект здобувача (здобувачки) освіти
відділення комп'ютерних систем

Мурзак Ірини Володимирівни

(прізвище, ім'я та по батькові)

Спеціальність: 123 "Комп'ютерна інженерія"

Освітня програма: «Комп'ютерна графіка і Web-дизайн»

Тема дипломного проекту: Розробка та впровадження адаптивного веб-сайту онлайн-навчання

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ

а) обсяг і якість виконання проекту (графічного матеріалу і розрахунково-пояснювальної записки) Дипломний проект виконано відповідно технічному завданню. Пояснювальна записка містить 72 сторінки. У пояснювальній записці виконано опис етапів розробки адаптивного веб-сайту онлайн навчання. Графічна частина складається з 14 слайдів мультимедійної презентації. Якість виконання пояснювальної записки та графічної частини відмінна, розробку виконано в повному обсязі.

б) самостійність роботи над проектом: Протягом всього строку дипломного проектування та переддипломної практики здобувачка освіти Мурзак І.В. поступово та послідовно виконувала всі етапи розробки. Всі роботи здобувачка освіти виконувала самостійно, з оглядом на рекомендації керівника.

в) теоретична підготовка випускника (випускниці): Здобувачка освіти Мурзак І.В. під час роботи над дипломним проектом вивчила достатню кількість літературних джерел та матеріалів за даною тематикою.

Вважаю, що теоретична підготовка дипломника добра і він готовий до захисту дипломного проекту.

г) вміння розв'язувати виробничі та конструкторські питання _____
Під час дипломного проектування здобувачка освіти Мурзак І.В. мала змогу
самостійно приймати окремі рішення з реалізації веб-сайту онлайн навчання
та показала вміння організовано працювати над поставленим завданням,
проектувати та розробляти веб-сайт за допомогою сучасного програмного
забезпечення, такого як Figma, Visual Studio Code, та сучасних фреймворків
для веб-розробки - Vue.js та Laravel.

Оцінка розрахункової частини _____	Відмінно _____
Оцінка графічної частини _____	Відмінно _____
Загальна оцінка _____	Відмінно _____

Прізвище, ім'я, по батькові керівника дипломного проекту _____
Кунуп Тетяна Василівна _____

Місце роботи і посада керівника дипломного проекту _____
ВСП "Одеський технічний фаховий коледж ОНТУ", викладач _____
спецдисциплін комісії комп'ютерних технологій та програмної інженерії _____

Підпис _____ 

« 09 » _____ 06. _____ 2023 р.

РЕЦЕНЗІЯ

на дипломний проект (роботу) здобувача (здобувачки) освіти
відділення комп'ютерних систем

Мурзак Ірини Володимирівни

(прізвище, ім'я та по батькові)

Спеціальність **123 «Комп'ютерна інженерія»**

Освітня програма **Комп'ютерна графіка та Web-дизайн**

Керівник дипломного проекту (роботи) **к.т.н., Кунуп Т.В.**

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи)

Розробка та впровадження адаптивного веб-сайту онлайн навчання

Обсяг розрахунково-пояснювальної записки 72 сторінок

Обсяг графічної (презентаційної) частини 14 аркушів (слайдів)

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту (роботи) завданню

Представлений на рецензію робота відповідає затверджений темі та виконаний відповідно технічному завданню.

б) характеристика виконання кожного розділу дипломного проекту (роботи)

Пояснювальна записка роботу виконана якісно, з дотриманням усіх норм та стандартів. В дипломному проекті проведено аналіз існуючих програмних рішень, розроблено веб-сайт, виконана інсталяція програмних залежностей, розроблена клієнтська частина веб-сайту виконано графічний дизайн сайту та проведено тестування сайту.

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту (роботи)

Графічна частина складається з 14 слайдів мультимедійної презентації, виконаної у програмному продукті MS PowerPoint, які містять ілюстративні слайди, передбачені технічним завданням. Пояснювальна записка виконана акуратно та у відповідності до норм. Якість виконання графічної частини проекту та пояснювальної записки висока, розробку виконано у повному обсязі

г) перелік позитивних якостей дипломного проекту (роботи) _____

У сучасну цифрову епоху веб-сайти та онлайн-платформи відіграють важливу роль у поширенні освітнього контенту та полегшенні засвоєння інформації. веб-сайт повинен мати простий і зрозумілий дизайн, та не містити занадто багато інформації Також певний інтерес представляє висновки та рекомендації щодо подальшого їх застосування під відповідні завдання для
д) основні недоліки дипломного проекту (роботи) _____

1. У пояснювальній записці не досить обоснована вибір програмного забезпечення для реалізації веб-сайту тобто, якому конкретному інструменту надано перевагу в розробці веб-сайту

2. У тексті пояснювальної записки зустрічаються друкарські помилки та неточності.

3. Відсутнє посилання літературу.

Оцінка розрахункової частини _____ *Відмінно*

Оцінка графічної частини _____ *Відмінно*

Загальна оцінка _____ *Відмінно*

Прізвище, ім'я, по батькові рецензента _____ Стайкуца Сергій Володимирович

Місце роботи і посада рецензента _____ Державний університет інтелектуальних технологій і зв'язку, к.ф.н., доцент кафедри КБ та ТЗІ, пом.декану факультету інформаційних технологій та кібербезпеки

Підпис: _____ *Стайкуца*
« 16 » _____ 2023 р.

ПІДПИС ПОСВІАЧЕНИЙ
НАЧАЛЬНИК ВІСЛА
КАДРІВ ДУІТЗ



Стайкуца

Ім'я користувача:
Наталія Вікторівна Копусь

ID перевірки:
1015227432

Дата перевірки:
24.05.2023 13:08:32 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
24.05.2023 13:08:49 EEST

ID користувача:
100011688

Назва документа: 4 КГ-06 Ірина Мурзак

Кількість сторінок: 55 Кількість слів: 6623 Кількість символів: 50038 Розмір файлу: 11.56 MB ID файлу: 1014903922

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

6.19%

Схожість

Найбільша схожість: 1.78% з Інтернет-джерелом (<https://er.nau.edu.ua/bitstream/NAU/57743/1/%d0%a4%d0%9a%d0%...>)

6.19% Джерела з Інтернету

70

Сторінка 57

Не знайдено джерел з Бібліотеки

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%

Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

1

Підозріле форматування

10
сторінок

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Мурзак Ірина Володимирівна,
здобувач освіти гр. 4КГ-06, та

Кунуп Тетяна Василівна,
керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до випускної кваліфікаційної роботи молодшого спеціаліста на тему:

**«Розробка та впровадження адаптивного веб-сайту онлайн навчання»
(автор роботи – Мурзак І.В., керівник роботи – Кунуп Т.В.)**

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2023 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи, і даємо згоду на обробку персональних даних.

Виконавець _____ / Мурзак І.В. /

Керівник _____ / Кунуп Т.В. /

« 09 » _____ 06 _____ 20 23 р.