

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «РОЗРОБКА ОНЛАЙН КУРСУ
«КОМП'ЮТЕРНА ГРАФІКА В WEB»

Виконала: студентка 4 курсу, групи 6.1219-2пі
спеціальності 121 інженерія програмного забезпечення
(шифр і назва спеціальності)
освітньої програми програмна інженерія
(назва освітньої програми)

Л.Р. Гасратова

(ініціали та прізвище)

Керівник доцент кафедри програмної інженерії,
доцент, к.т.н. Мухін В.В.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент завідувач кафедри фундаментальної та прикладної
математики, професор, д.т.н. Гребенюк С.М.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний
Кафедра програмної інженерії
Рівень вищої освіти бакалавр
Спеціальність 121 інженерія програмного забезпечення
(шифр і назва)
Освітня програма програмна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри програмної
інженерії, к.ф.-м.н., доцент

_____ Лісняк А.О.
(підпис)

“ 07 ” 02 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ

Гасратовій Лейлі Русланівні

(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка онлайн курсу «Комп'ютерна графіка в Web»

керівник роботи Мухін Віталій Вікторович, к.т.н., доцент

(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 26 » січня 2023 року № 102-с

2. Строк подання студентом роботи 07.06.2023 р.

3. Вихідні дані до роботи 1. Постановка задачі.

2. Перелік літератури.

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

1. Постановка задачі, аналіз предметної області.

2. Аналіз вимог до інформаційної системи.

3. Реалізація інформаційної системи.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____

презентація


6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 07.02.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	08.02.2023	
2.	Збір вихідних даних.	20.02.2023	
3.	Обробка методичних та теоретичних джерел.	06.03.2023	
4.	Розробка першого та другого розділу.	03.04.2023	
5.	Розробка третього розділу та четвертого розділу.	15.05.2023	
6.	Оформлення та нормоконтроль кваліфікаційної роботи.	01.06.2023	
7.	Захист кваліфікаційної роботи.	22.06.2023	

Студент 
(підпис)

Л.Р. Гасратова
(ініціали та прізвище)

Керівник роботи
(підпис)

В.В. Мухін
(ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер
(підпис)

А.В. Столярова
(ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка онлайн курсу «Комп'ютерна графіка в Web»: 46 с., 34 рис., 1 табл., 9 джерел.

БРАУЗЕР, ВЕБ-ДОДАТОК, ІНФОРМАЦІЙНІ СИСТЕМИ, КУРСИ, САЙТ, CSS, JAVASCRIPT, HTML, MONGODB, REACT, WEB-КУРСИ.

Об'єкт дослідження – розробка та проектування веб-сайту, спрямованого на надання онлайн курсу з комп'ютерної графіки в Web.

Мета роботи: створення функціонального та ефективного веб-додатку, який надаватиме користувачам можливість навчатися та отримувати інформацію про комп'ютерну графіку в онлайн форматі.

Методи дослідження: аналіз предметної області, вивчення та узагальнення, моделювання.

У кваліфікаційній роботі досліджено предметну область, засоби реалізації веб-додатку для онлайн-курсів. Розглянуто основні особливості JS, React, реалізовано базу даних.

SUMMARY

Bachelor's qualifying paper «Development of the Online Course «Computer Graphics on the Web»: 46 pages, 34 figures, 1 table, 9 references.

BROWSER, WEB APPLICATION, INFORMATION SYSTEMS, COURSES, WEBSITE, CSS, JAVASCRIPT, HTML, MONGODB, REACT, WEB COURSES.

Object of the study: development and design of a website aimed at providing an online course on computer graphics in Web.

Aim of the study: create a functional and efficient web application that will provide users with the opportunity to learn and obtain information about computer graphics in an online format.

Methods of research: analysis of the subject area, summary and generalization, modeling.

The qualification work explores the subject area and the tools for implementing a web application for online courses. The main features of JS and React were considered and a database is implemented.

ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат	4
Summary	5
Вступ.....	7
1 Аналіз предметної області.....	9
1.1 Опис предметної області	9
1.2 Технічне завдання	11
1.3 Огляд існуючих аналогів	12
2 Проєктування програмного забезпечення	14
2.1 JavaScript	14
2.2 Фреймворки/бібліотеки JavaScript	16
2.3 Діаграма прецедентів	21
3 Розробка веб-додатку	23
3.1 Розробка дизайну	23
3.2 Файлова структура проєкту	25
3.3 Розробка веб-сайту	28
3.4 База даних	33
3.5 Реалізація бази даних	35
4 Приклад роботи додатку	38
Висновки	45
Перелік посилань.....	46

ВСТУП

В сучасному світі, де комп'ютерна графіка використовується майже в усіх сферах життя, навчитися створювати графічні елементи для веб-сайтів є надзвичайно важливою навичкою. Із зростанням кількості інтернет-користувачів, веб-дизайн та веб-розробка стали популярними професіями. Однак, без навичок комп'ютерної графіки, важко досягти успіху у цих галузях.

Онлайн курси з комп'ютерної графіки веб-дизайну є дуже важливими в даний момент по кількох причинах:

- розвиток інтернету та електронної комунікації: завдяки швидкому зростанню інтернет-користувачів та електронних майданчиків, веб-розробка та веб-дизайн стали дуже важливими для бізнесу та інтернет-комунікації;

- вимоги до веб-дизайнерів та розробників: зараз веб-дизайнерам та розробникам потрібно мати широкі знання та навички в галузі комп'ютерної графіки, щоб створювати якісні веб-сайти та додатки, які відповідають сучасним стандартам;

- онлайн навчання: завдяки зростанню популярності онлайн-навчання, зручність та доступність навчальних курсів з комп'ютерної графіки веб-дизайну зросли; люди можуть навчатися в будь-який зручний для них час та місце, а також отримувати якісну освіту від досвідчених фахівців з усього світу;

- розвиток технологій: завдяки постійному розвитку технологій, з'являються нові інструменти та програмні засоби для створення веб-сайтів та додатків, що вимагає від веб-дизайнерів та розробників постійного оновлення своїх знань та навичок.

Тому, розробка онлайн курсу «Комп'ютерна графіка в Web» може допомогти студентам та веб-розробникам отримати необхідні знання та навички для створення професійних графічних елементів для веб-сайтів. У цьому курсі студенти зможуть ознайомитися з основами дизайну, вивчити роботу з

графічними програмами, оволодіти техніками створення векторної та растрової графіки, а також навчитися працювати з анімацією та відео.

Розробка такого курсу допоможе студентам та професіоналам розширити свої знання та навички, що в свою чергу, дозволить їм створювати візуально привабливі та ефективні веб-сайти та додатки.

Метою роботи є розробка веб-сайту на тему онлайн курсу «Комп'ютерна графіка в Web». Тобто спроектувати, розробити та протестувати роботу веб-додатку.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області

З появою Інтернету вектор розвитку нашої цивілізації змінився остаточно. Ще буквально 30 років назад люди витрачали велику кількість часу, сил та грошей, щоб, наприклад, елементарно написати листа чи відправити телеграму, що в наш час являється невід'ємною частиною нашої буденності і забирає у нас лише декілька секунд. Така технологія, як Інтернет, була запланована як легкий та швидкий спосіб комунікації в теперішніх реаліях відіграє набагато більшу роль, ніж можна уявити.

Перше враження, що приходить після слова «Інтернет» – це абсолютно новий світ, інший, альтернативний портативний всесвіт, в якому є практично все, що потрібно середньостатистичній людині. Соціальні мережі в силах замінити для нас реальне спілкування, online-робота на відстані або так звані «підроботки» в Інтернеті можуть певною мірою забезпечити безбідне існування тим, хто розуміється на цьому, online-магазини взагалі змінили тогочасний ринок, вивівши його на новий рівень: продукти за рахунок великої конкуренції мають вищу якість та помірну ціну; продаж продуктів, що є доцільними та актуальними в межах певної країни та заданого там тренду призводить до активного і стрімкого розвитку малого бізнесу, що вкрай позитивно впливає на економіку та, власне, на ВВП цієї країни; завдяки поширенню мультимедійної сфери та сфери розваг виникає велика кількість товарів популярних франшиз, що задовольняють емоційний та особистісний аспект людини, яка купує цей товар.

Але зовсім недавно можливості Інтернет розширились та грубо кажучи «зайшли на територію педагогіки». Іще донедавна навчання будь-чому вважалось надважкою працею, що потребує зазубрювання, концентрації, витримки та швидкості в розумінні, адже якщо дитина чи молода людина щось прогавить на своїх заняттях, які йдуть за однією й тією ж парадигмою, то

зрозуміти наступний матеріал їй уже може бути складно. Пандемія ХХ-го століття наочно показала нам, що можна використовувати Інтернет не лише для розваг, а й для розширення власних кордонів, для вивчення чогось нового, для пошуку цікавих хобі, які в результаті можуть стати повноцінною роботою. Якщо раніше людина обирала одну професію на все життя і не могла змінити свій вибір через брак коштів, або тому, що вивчення на конкретну посаду займало від 4-х років, то зараз за невелику суму грошей можна купляти онлайн-уроки, вебінари, онлайн-курси та репетиторства. Тепер звичайна людина, яка розуміється, наприклад, лише на математичних науках чи медицині, купуючи онлайн-курси може навчитись професійно робити макіяж, або оволодіти ІТ-технологіям.

У 2023 році онлайн-курси досягли свого найбільшого розвитку, адже кожен популярний блогер, який має принаймні мінімальну кількість фоловерів здатних купувати продукти, створює свої власні курси на основі набутого досвіду, наприклад, курси для початківців в малому бізнес, курси по вивченню іноземних мов та ін.. Блогери, які ведуть сторінку для висвітлення свого роду діяльності, роблять курси для оволодіння певної професії, або набуття кваліфікації. Це можуть бути курси для схуднення, засвоєння графічного дизайну, вивчення вокалу та гри на різних музичних інструментів, пошив одягу, іграшок, розробка онлайн-ігор, курси для програмістів та багато іншого.

Особлива увага приділяється курсам для молоді, що має чисто педагогічне спрямування, наприклад, для покращення знань з якоїсь певної дисципліни: лінгвістики, математики, інженерія, бізнес, суспільні науки, психологія, інформатика. Існує велика мережа іноземних сайтів, які допомагають студентам в цьому. Для прикладу можу привести Coursera – це глобальна платформа онлайн-навчання, яка пропонує будь-кому та будь-де доступ до онлайн-курсів та дипломів від провідних університетів та компаній [1].

Студенти здобувають навички, необхідні для працевлаштування за отриманою спеціальністю, а також доступ до ресурсів, які дозволяють претендувати на відповідні робочі місця.

Отже, онлайн-курси – це майбутня форма навчання, де є доступ до дуже різноманітних матеріалів, таких як відео, фотографії та електронні книги, і цей додатковий вміст доступний у будь-який момент з будь-якого місця, що запропонує вам більш динамічну та індивідуальну освіту.

1.2 Технічне завдання

ТЗ – основний документ, яким керуються розробники та проєктувальники, розпочинаючи розробку нового виробу. Воно визначає основні напрямки розробки – конструкції та принципи роботи майбутнього виробу. ТЗ заявляє, з одного боку, про потреби суспільства у нових виробах, з іншого – про технічні та техніко-економічні характеристики виробу.

Для коректної роботи сайту потрібні певні критерії, які обирає замовник або розробник сайту.

Для свого сайту я обрала такі функціональні вимоги до сайту:

- 1) авторизація для збереження конфіденційності та цілісності інформації в системі, для захисту акаунтів користувачів та їх персональних даних;
- 2) перегляд всіх доступних курсів (JS, HTML, CSS);
- 3) інтуїтивно зрозумілий дизайн;
- 4) редагування своїх даних, видалення свого акаунту;
- 5) передбачити можливість невірно введених даних;
- 6) можливість онлайн тестування коду, який буде на онлайн-уроці.

Отже, ТЗ цієї роботи полягає в наступному: розробити сайт, призначений для онлайн курсів з WEB-дизайну, вивчити предметну область, врахувати переваги та недоліки існуючих ринкових рішень.

1.3 Огляд існуючих аналогів

Для того, щоб достовірно зрозуміти потрібні функції були розглянуті такі аналоги: HTML5 & CSS3 Fundamentals, Ultimate Web Design Course.

HTML5 & CSS3 Fundamentals – це онлайн-курс від компанії Microsoft. Це курс, який протягом 21 епізоду (рис. 1.1) Боб Табор з www.LearnVisualStudio.net навчить основам програмування HTML5 та CSS3. Там розглядається концепції веб-сторінок, стилів CSS3 та можливостей HTML5.

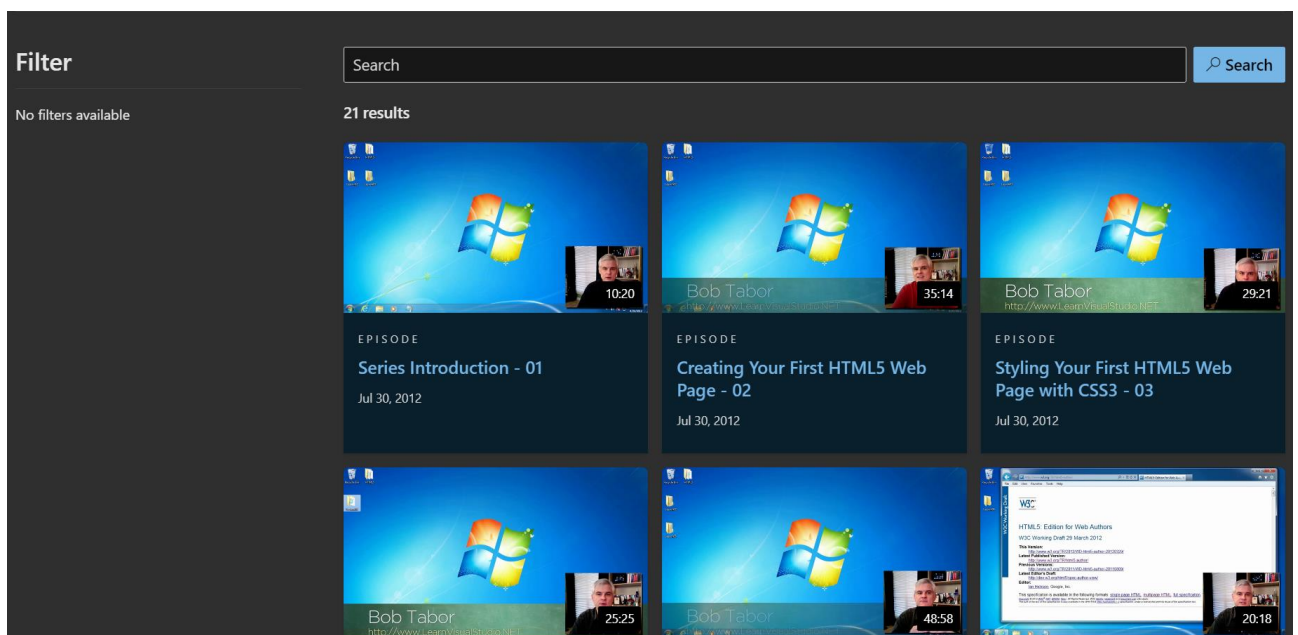


Рисунок 1.1 – Онлайн курси від компанії Microsoft

Основна перевага цих уроків полягає в їх доступності тому, що вони завантажені на сайт і при бажанні їх можна переглянути. Але попри таку значну перевагу існує ряд недоліків. Ось декілька з них:

- 1) відсутність сертифікату про проходження курсу;
- 2) відсутність взаємозв'язку з інструктором (питання та відповіді інструктора).

Webflow University – це абсолютно безкоштовна платформа, яка пропонує комплексні курси та підручники з веб-дизайну. Курс Ultimate Web Design має комплексну навчальну програму, яка охоплює все: від базових HTML і CSS до

3D-дизайну, типографіки та SEO. Цей курс, що триває лише п'ять годин (рис. 1.2), дає змогу учасникам швидко створити фундамент у веб-дизайні. Курс також навчить створювати професійні веб-сайти без кодування за допомогою платформи Webflow [2].

Summary

The Ultimate web design course includes over 100 videos. It starts with everything you need to know to get started with Webflow and eases into more advanced design concepts. You'll learn the basics of HTML, how CSS classes and subclasses function, how to construct a page design with elements like containers, divs, flexboxes, and grids. And of course, we'll take you through how to use one of our most powerful features – Webflow CMS. By the time you finish this course, you'll be able to build some pretty amazing stuff.

Course outline

Getting started	
Intro to HTML	5:17
HTML structure	1:55
Web structure	
Intro to the box model	3:04
Navigator panel	2:45
Elements	
Use text columns in Webflow	
Buttons & links	
Navbar menu button	4:51
Link block	9:35

Рисунок 1.2 – Webflow University, курс Ultimate Web Design

Серед переваг цього курсу можна виділити:

- 1) безкоштовний курс;
- 2) гнучкий графік.

А що стосується недоліків, знову ж таки відсутність сертифікату.

Отже, було розглянуто декілька онлайн курсів; більша частина курсів розроблена за типом відео-уроків. Але варто враховувати, що не кожен зможе дивитися відео, тому я обрала такий варіант, як урок текстом. На мою думку, це допоможе суб'єкту краще запам'ятовувати інформацію, а також додавати приклади коду з можливістю їх відтворення, адже найголовніше – практика.

2 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 JavaScript

JavaScript – це мова сценаріїв для створення динамічного вмісту веб-сторінок.

JavaScript з'явився у 1995 році як спосіб додавання програм до веб-сторінок у браузері Netscape Navigator. З тих пір мова була прийнята всіма іншими основними графічними веб-браузерами. Завдяки їй стали можливими сучасні веб-додатки – додатки, з якими можна взаємодіяти безпосередньо, не перезавантажуючи сторінку для кожної дії. JavaScript також використовується на більш традиційних веб-сайтах для забезпечення різних форм інтерактивності та інтелектуальності [3].

Цю мову можна використовувати в багатьох сферах, наприклад, front-end, створення мобільного ПО, back-end, написання програмного забезпечення десктопного типу.

Наразі JavaScript неймовірно універсальний. Будь-який браузер, будь-який комп'ютерний пристрій читає код на JavaScript, а зараз навіть мобільні додатки. Вихідні коди передаються і виконуються у вихідному вигляді, можна запросто витягти скрипт з якогось сайту, підправити і забрати собі. За допомогою JS можна почати з малого: з каруселей, слайдів зображень, макетів, що змінюються, і відповідей на натискання кнопки. Але з великим досвідом є можливість створювати ігри, анімовані 2D і 3D графіки, комплексні додатки на базі даних, і багато іншого. На квітень 2023 року, JS займає 3 місце, продемонстровано на рисунку 2.1, у PYPL (Popularity of Programming Language).

Це зумовлено тим, що JS ідеально підходить для back-програмування. Кодування на JavaScript вважається однією з найпростіших у підтримці мов веб-розробки. Крім того, розробка на JavaScript є економічно вигідною.

Rank	Change	Language	Share	Trend
1		Python	27.43 %	-0.8 %
2		Java	16.41 %	-1.7 %
3		JavaScript	9.57 %	+0.3 %
4		C#	6.9 %	-0.3 %
5		C/C++	6.65 %	-0.5 %
6		PHP	5.17 %	-0.5 %
7		R	4.22 %	-0.4 %
8		TypeScript	2.89 %	+0.5 %
9	↑	Swift	2.31 %	+0.2 %
10	↓	Objective-C	2.09 %	-0.1 %

Рисунок 2.1 – Рейтинг Popularity of Programming Language

Але варто розуміти, що яким би популярним та функціональним не був JavaScript серед розробників, в нього як і в кожній мові програмування є свої переваги і недоліки.

Насамперед розглянемо такі переваги:

- незамінність у веб-розробці. JS – це основна «технологія» для клієнт-серверних моделей і програмування «для інтернету»;
- наявність повної інтеграції з версткою сторінок і серверною частиною;
- раціональність застосування і простота;
- наявність власної потужної екосистеми (інфраструктури).

Існують також такі недоліки:

- мова компілюється в момент виконання коду, що збільшує час виконання програми;
- відсутність типізації даних і тд.

Отже, JS – це ідеальний варіант для написання інтерактивних веб-сторінок, анімаційних ефектів. Завдяки масштабній популяризації у світі і враховуючи те,

що JS займає перше місце в рейтингу Popularity of Programming Language, я буду використовувати у своїй дипломній роботі.

2.2 Фреймворки/бібліотеки JavaScript

Сьогодні front-end-фреймворки та бібліотеки стають невід’ємною частиною сучасного стеку веб-розробки. Їх існує дуже багато, але є найпоширеніші бібліотеки/фреймворки, які зображені на рисунку 2.2 [4].



Рисунок 2.2 – Фреймворки JS

При виборі фреймворку, потрібно звертати увагу на:

- розмір спільноти;
- стійкість;
- підтримка;
- безпека;
- документація.

React.js – це фреймворк і бібліотека JavaScript з відкритим вихідним кодом, розроблена компанією Facebook. Він використовується для швидкого та ефективного створення інтерактивних користувацьких інтерфейсів і веб-додатків зі значно меншою кількістю коду, ніж у звичайному JavaScript [5].

У React розробка додатків, створюючи багаторазові компоненти, уявляється незалежними блоками Lego. Ці компоненти є окремими частинами кінцевого інтерфейсу, які будучи зібраними формують весь користувацький інтерфейс додатку.

Основними перевагами цього фреймворку/бібліотеки є те, що React змінює вміст сторінки без її ж перезавантаження. Завдяки цьому взаємодія додатка і користувача є швидшою. Також React використовує віртуальний Document Object Model (DOM). Браузер, коли здійснює запит та отримує відповідь від сервера, її вихідний HTML код, повинен його розібрати, тобто він переглядає та створює приблизно таке дерево, яке зображено на рисунку 2.3

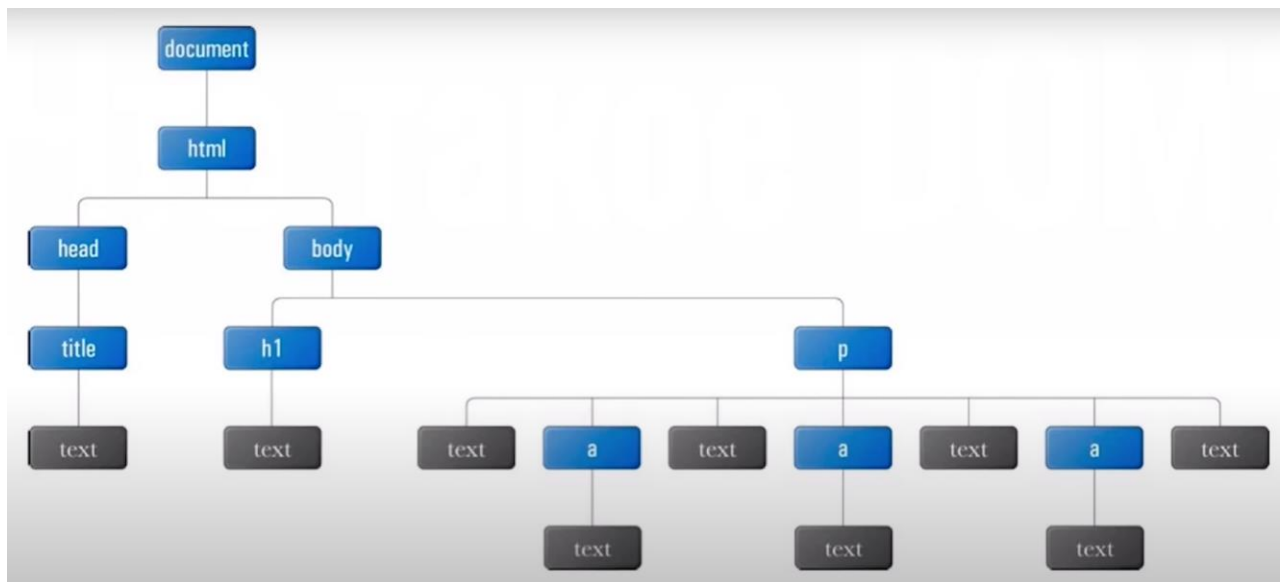
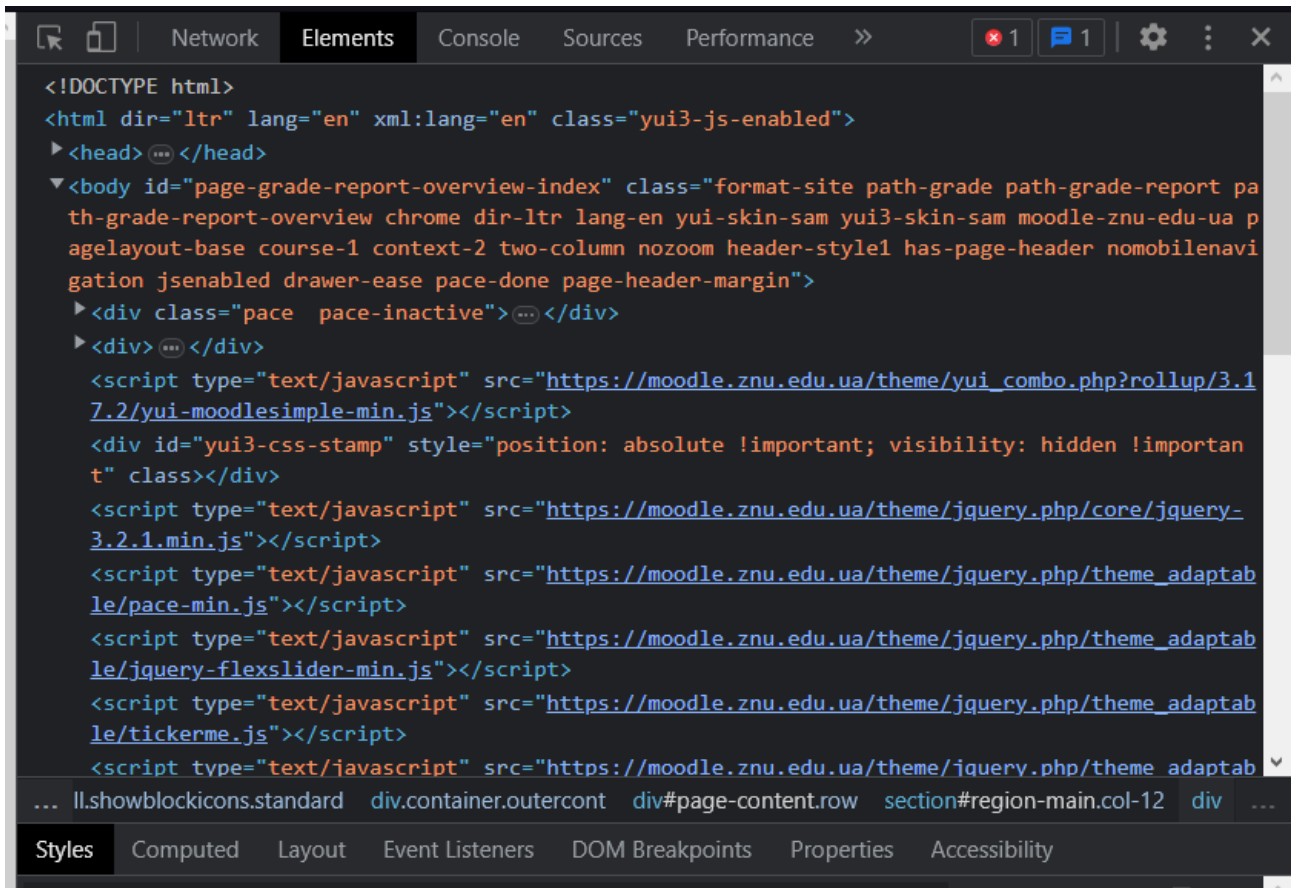


Рисунок 2.3 – Дерево HTML

Іншими словами, він складає елементи один з одним у правильній послідовності, а для перегляду DOM можна скористатись переглядом коду елемента (див. рис. 2.4).

Це об'єкт у якому збережена інформація про стан інтерфейсу та саме він допомагає отримувати доступ до елементів сторінки за сприянням JavaScript. Але, коли є потреба маніпулювати HTML елементами, можна зіткнутись зі зниженням продуктивності, особливо, коли цих елементів багато. Для вирішення цієї проблеми з'явилась концепція віртуального DOM-у.



```

<!DOCTYPE html>
<html dir="ltr" lang="en" xml:lang="en" class="yui3-js-enabled">
  <head> </head>
  <body id="page-grade-report-overview-index" class="format-site path-grade path-grade-report path-grade-report-overview chrome dir-ltr lang-en yui-skin-sam yui3-skin-sam moodle-znu-edu-ua page-layout-base course-1 context-2 two-column nozoom header-style1 has-page-header nomobilenavigation jsenabled drawer-ease pace-done page-header-margin">
    <div class="pace pace-inactive"> </div>
    <div> </div>
    <script type="text/javascript" src="https://moodle.znu.edu.ua/theme/yui_combo.php?rollup/3.17.2/yui-moodlesimple-min.js"></script>
    <div id="yui3-css-stamp" style="position: absolute !important; visibility: hidden !important" class></div>
    <script type="text/javascript" src="https://moodle.znu.edu.ua/theme/jquery.php/core/jquery-3.2.1.min.js"></script>
    <script type="text/javascript" src="https://moodle.znu.edu.ua/theme/jquery.php/theme_adaptable/pace-min.js"></script>
    <script type="text/javascript" src="https://moodle.znu.edu.ua/theme/jquery.php/theme_adaptable/jquery-flexslider-min.js"></script>
    <script type="text/javascript" src="https://moodle.znu.edu.ua/theme/jquery.php/theme_adaptable/tickerme.js"></script>
    <script type="text/javascript" src="https://moodle.znu.edu.ua/theme/jquery.php/theme_adaptable/showblockicons.standard"></script>
  </body>

```

Рисунок 2.4 – Перегляд коду елемента

Віртуальний DOM представляє легковажну копію звичайного DOM-у. Якщо необхідно змінити елементи веб-сторінки, то зміни спочатку вносяться у віртуальну версію, потім новий стан віртуального DOM-у порівнюється з поточним, і якщо ці стани відрізняються, то React знаходить мінімальну кількість маніпуляцій, які потрібні для поновлення реального DOM-у до нового стану, та виконує їх. У підсумку така схема роботи з елементами веб-сторінки працює набагато швидше й ефективніше.

Angular – це JavaScript фреймворк, створений компанією Google, побудована на TypeScript. Angular включає в себе:

- компонентний фреймворк для створення масштабованих веб-додатків;
- набір інтегрованих бібліотек, які охоплюють широкий спектр функцій, включаючи маршрутизацію, управління формами, клієнт-серверну взаємодію і т.д.;
- набір інструментів для розробників, які допоможуть розробляти, збирати, тестувати та оновлювати код [5].

Фреймворк дає змогу створювати не тільки веб-додатки. З його допомогою можна писати код, який може бути адаптований під інше середовище. Наприклад, додаток зможе працювати в мобільній або десктопній операційній системі. За допомогою Angular можна створити навіть додаток для доповненої реальності.

Angular, на відмінно від свого конкурента React, має інкрементний DOM. Інкрементний DOM зберігає канонічне подання DOM у самому собі. Іншими словами, в процесі не використовується додаткова пам'ять, що має значення для пристроїв з обмеженим обсягом пам'яті.

Таким чином, додатки на Angular виграють від меншого споживання пам'яті, в той час як React кращий у продуктивності. При цьому інкрементний DOM все одно може давати швидкість понад 60 кадрів за секунду, що означає, що різниця в продуктивності не буде помітна звичайному користувачеві. Більш детальне порівняння зображено у таблиці 2.1. Крім продуктивності та споживання пам'яті, інші фактори, такі як екосистема, розширюваність, спільнота розробників і наявність інструментів та бібліотек, також можуть вплинути на вибір між Angular та React. Однак, вибір між Angular та React залежить від конкретних вимог і потреб проєкту, тому важливо ретельно проаналізувати їх особливості перед прийняттям рішення.

Таблиця 2.1 – Порівняння Angular та React

Характеристика	Angular	React
Призначення	Повнофункціональна платформа – дає чітке уявлення про те, як має бути спроектований застосунок, а також містить у собі низку критичних бібліотек	Бібліотека займається тільки компонентами користувацького інтерфейсу
DOM	Інкрементний	Віртуальний
Можливості UI	Містить низку компонентів Material UI, які спрощують налаштування користувацького інтерфейсу	Розроблені спільнотою інструменти користувацького інтерфейсу надають широкий спектр компонентів із можливістю їхньої кастомізації
Прив'язка даних	Одностороння і двостороння	Одностороння прив'язка
Для яких додатків підходить	Складних додатків з великим навантаженням	Додатки будь-якого обсягу і складності

Отже, на основі розглянутого вище матеріалу, на мою думку у моїй роботі доречно буде використати React, адже він найпопулярніший серед фреймворків/бібліотек JS, адже в ньому присутні такі переваги, яких немає у React аналогів. Ці переваги гарно синхронізується з моєю дипломною роботою.

2.3 Діаграма прецедентів

Діаграма прецедентів – це одна з ключових діаграм, використовуваних при проектуванні програмного забезпечення. Вона зображує взаємодії між системою (або програмою) та її користувачами, які називаються акторами [6].

Діаграма прецедентів складається з акторів, прецедентів та взаємодій між ними. Актори – це користувачі системи або інші системи, які взаємодіють з нею. Прецеденти – це опис дії, яку може виконати система або її частина. Взаємодії – це способи, за якими актор взаємодіє з системою.

Діаграма прецедентів може бути корисна при проектуванні програмного забезпечення, оскільки вона допомагає зрозуміти вимоги користувачів та функціональність системи. Вона також може використовуватися для комунікації між розробниками та клієнтами, оскільки вона надає чітке візуальне представлення того, як буде використовуватися програмне забезпечення. На рисунку 2.4. зображена діаграма мого проекту.

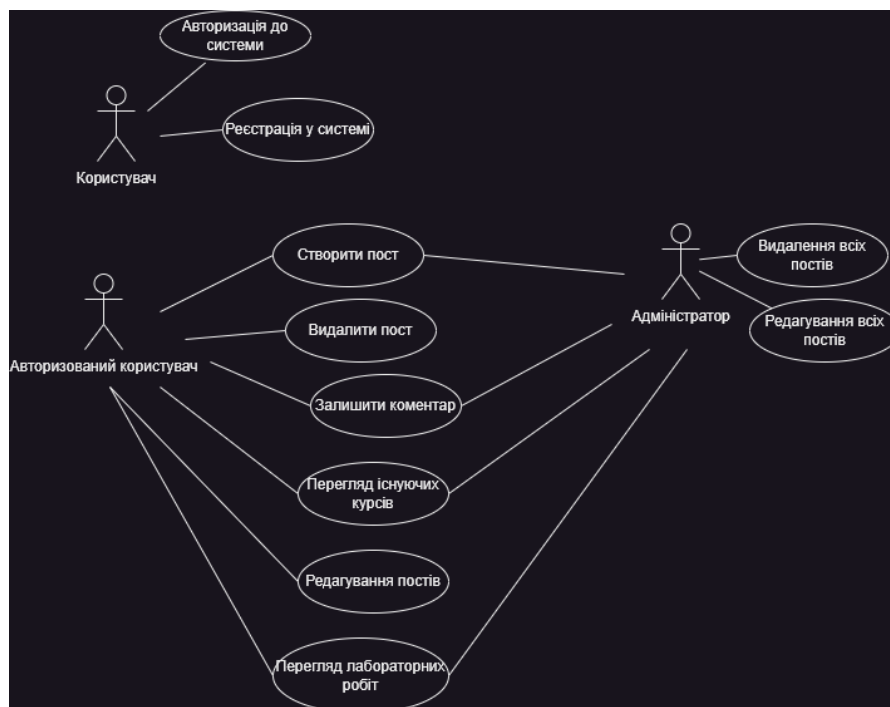


Рисунок 2.5 – Діаграма прецедентів

Отже, діаграма прецедентів допомагає розібратися в тому, як система буде взаємодіяти з користувачами та які є їх потреби. Вона дозволяє виявити основні функції, які потрібні користувачам та які можуть бути необхідними для роботи системи. Також діаграма допомагає зрозуміти, як елементи системи взаємодіють між собою та з користувачами. Це дозволяє підтримувати правильну архітектуру системи.

3 РОЗРОБКА ВЕБ-ДОДАТКУ

3.1 Розробка дизайну

Дизайн сайту – це одна з найголовніших частин у проєкті, адже він може як відштовхувати, так і приваблювати нових користувачів.

Дизайн сайту є надзвичайно важливим з кількох причин, розглянемо їх нижче.

Візуальне враження. Перше, що бачать люди, коли заходять на сайт – зовнішній вигляд. Хороший дизайн створює гарне враження у користувачів.

Зручність у використанні. Якісний дизайн спрощує навігацію сайтом і робить його інтуїтивно зрозумілим. Користувачі зможуть швидко знайти потрібну інформацію, що збільшить їхнє задоволення від використання сайту.

Унікальність та індивідуальність. Відмінний дизайн сайту допомагає виділитися серед конкурентів і змушує відвідувачів запам'ятати ваш бренд.

Для того, щоб задовольнити смаки кожного при створенні дизайну, було обрано мінімалізм як оптимальний стиль.

Мінімалізм (у WEB-дизайні) – це у веб-дизайні означає спрощення інтерфейсу шляхом видалення зайвих елементів. Іншими словами, для досягнення більшого результату за допомогою обмежених ресурсів [7].

Веб-додаток створений у темних тонах та з мінімалістичним дизайном. На рисунку 3.1 зображена головна сторінка сайту.



Рисунок 3.1 – Головна сторінка сайту

Стиль курсів зображений на рис. 3.2.

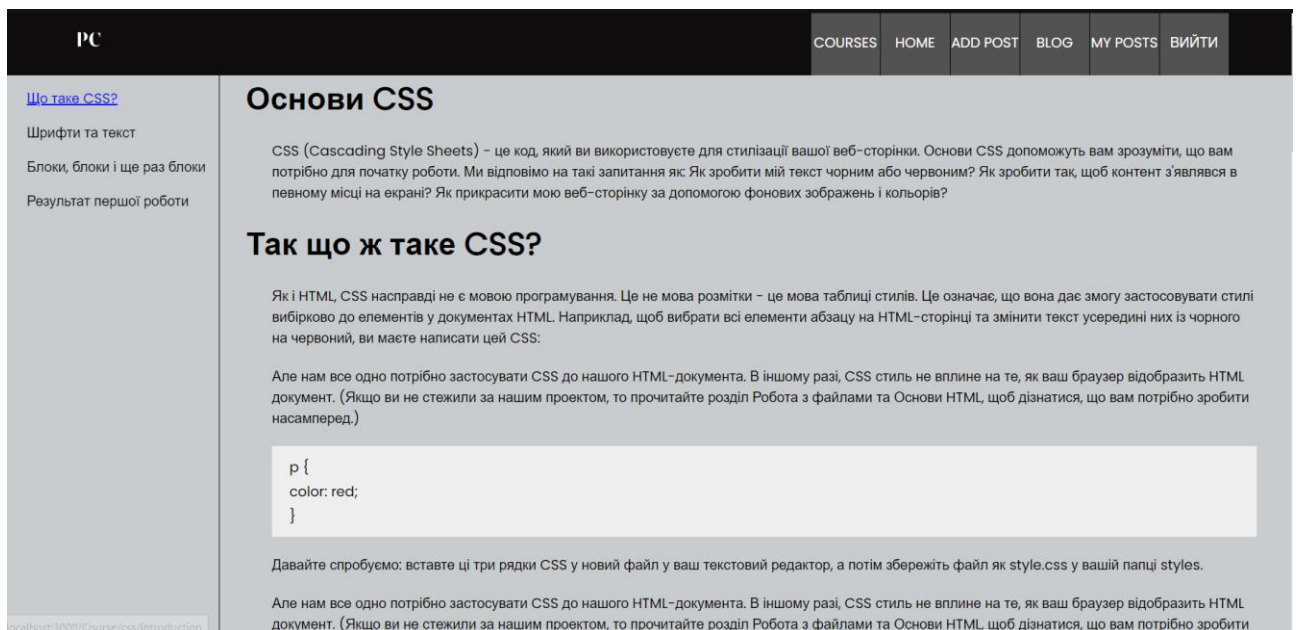


Рисунок 3.2 – Стиль сторінки курсів

Програмування комп'ютерної графіки у Веб – один із вибіркових курсів університету ЗНУ з підготовки фахівця з програмної інженерії. Курс має на меті засвоєння студентами технології побудови тривимірної графіки в браузері. з використанням 3D бібліотек JavaScript (Three.js, Tween.js), конструктивної

твердотільної геометрії (ThreeCSG.js). На рисунку 3.3 зображено перелік лабораторних робіт з цього курсу, де є можливість завантажити код, переглянути завдання та подивитись на результат [8].

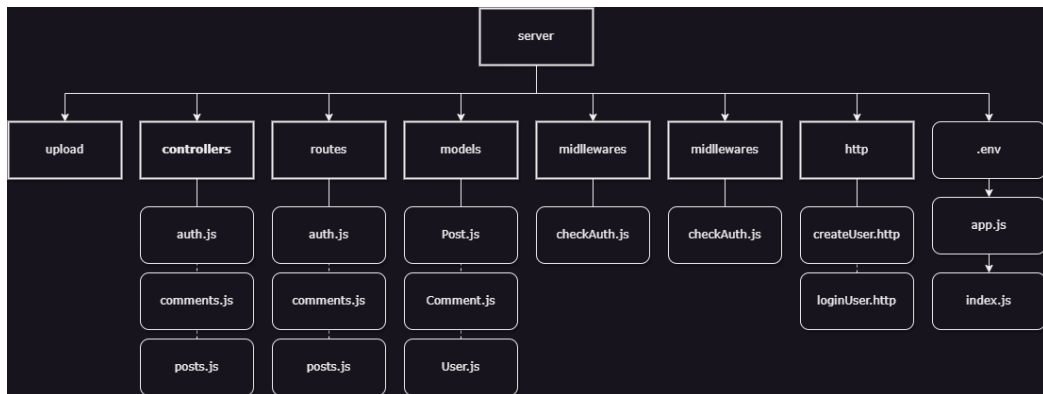


Рисунок 3.3 – Лабораторні роботи

Отже, дизайн створений у мінімалістичному стилі, що забезпечує простоту та лаконічність у розташуванні елементів на сайті. Це полегшує користувачам орієнтуватись та швидко знаходити необхідну інформацію. Також мінімалістичний дизайн зазвичай має менше графічних та мультимедійних елементів, що допомагає сайту завантажуватись швидше, допомагає підвищувати рівень довіри до сайту та його власника, оскільки він виглядає більш професійним та дбайливо продуманим.

3.2 Файлова структура проєкту

Фреймворк/бібліотека React має свою архітектуру файлів проєкту. Один із популярних підходів – розміщення файлів CSS, JS і тестів у папках, згрупованих за функціональністю або маршрутом.

Файлова структура серверної частини проєкту зображена на рисунку 3.4.

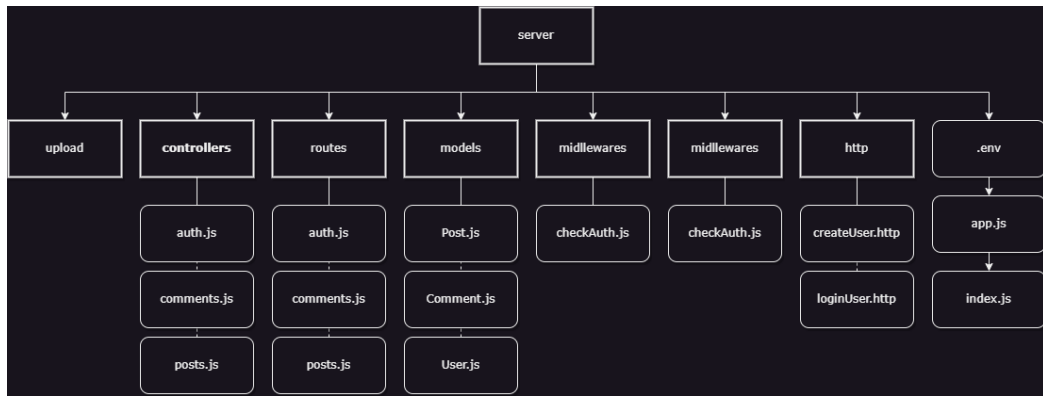


Рисунок 3.4 – Файлова структура серверної частини проекту

В серверній папці знаходяться такі директорії:

- upload містить в собі зображення з блогів користувачів;
- routes містить в собі файли, які допомагають в маршрутизації, переходити на сторінки без перезавантаження;
- models містить в собі файли, для взаємодії з базою даних;
- middlewares містить в собі файл, для проміжних функцій, що мають доступ до об'єкта запиту (req), об'єкта відповіді (res) і до наступної функції проміжної обробки в циклі «запит-відповідь» застосунку.

Файлова структура клієнтської частини проекту зображена на рисунку 3.5.

public	5/29/2023 6:51 PM	File folder	
src	5/29/2023 6:51 PM	File folder	
.gitignore	5/29/2023 6:51 PM	Git Ignore Source ...	1 KB
package	5/29/2023 6:51 PM	JSON Source File	2 KB
package-lock	5/29/2023 6:58 PM	JSON Source File	672 KB
README	5/29/2023 6:51 PM	Markdown Source ...	4 KB

Рисунок 3.5 – Файлова структура клієнтської частини проекту

У папці public зберігаються зображення, відео, шрифти, які використовуються у клієнтській частині проекту.

src – це основна папка клієнтської частини. Вміст директорії src зображений на рисунку 3.6.

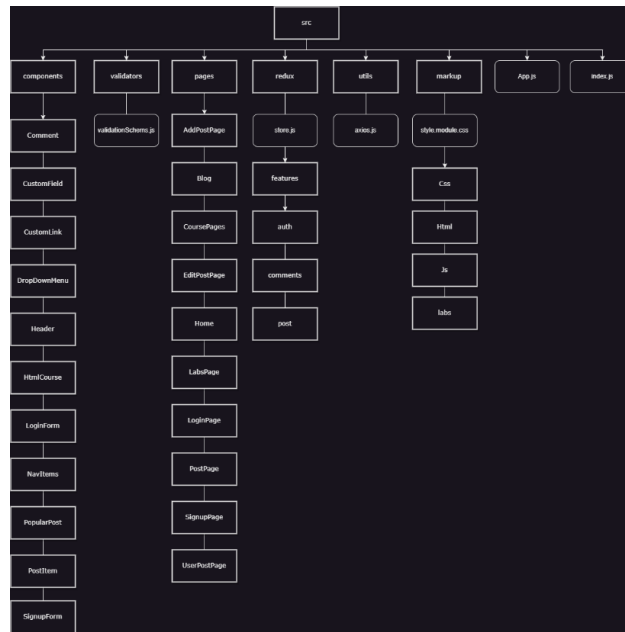


Рисунок 3.6 – Директорія src

У папці на знаходяться наступні каталоги:

- markup містить в собі всі сторінки курсів, лабораторних робіт, та їх css-файли;
- pages містить сторінки всього проєкту: додавання посту, блог, курси (меню зліва), редагування посту, домашня сторінка, сторінка логіну та реєстрації, сторінка постів, пости користувача;
- redux – тут знаходяться файли, в яких використовується ця бібліотека, а саме авторизація, пости та коментарі, що спрощує керування станом, полегшує відлагодження та тестування, сприяє розширюваності та підтримує розробку складних додатків;
- validators містить в собі один файл, де дані перевіряються на валідність.

У процесі роботи з проєктом, папка «node_modules» автоматично створюється під час встановлення залежностей, які вказані у файлі «package.json». Ця папка містить всі залежності, такі як бібліотеки та пакети, які використовуються у проєкті. Вона знаходиться у двох директоріях: одна в серверній частині проєкту, а інша – у клієнтській частині.

3.3 Розробка веб-сайту

Розробка веб-сайту за допомогою React виконується шляхом створення компонентів, які відповідають за різні частини і функціональність сайту. Кожен компонент містить свою логіку, стиль та може містити вкладені компоненти. React забезпечує декларативний підхід до розробки, де стан компонентів оновлюється автоматично при зміні даних, що сприяє більш простій та ефективній розробці.

За допомогою наведеного на рисунку 3.7 коду було створено головну сторінку веб-сайту. Цей код містить необхідні елементи та компоненти, які відображають контент та функціональність сторінки. Шляхом використання HTML, CSS та React-специфічних елементів та властивостей, розробники зуміли створити вигляд та поведінку головної сторінки, забезпечуючи зручну навігацію та доступ до важливої інформації для користувачів. Загальний результат роботи коду полягає в створенні привабливої та функціональної головної сторінки, яка ефективно взаємодіє з користувачем та задовольняє їх потреби.

```

client > src > pages > Home > JS Homejs > Home
1 import React from 'react';
2 import Header from '../components/Header/Header';
3 import styles from './Home.module.css';
4 import { Link } from 'react-router-dom';
5 import DropdownMenu from '../components/DropDownMenu/DropDownMenu';
6 import { navItems } from '../components/NavItems/NavItems';
7 import { useSelector } from 'react-redux';
8 import { checkAuth } from '../redux/features/auth/authSlice';
9 //import img from '../img/title.jpg'
10
11 console.log(navItems);
12
13 export default function Home() {
14   const isAuth = useSelector(checkAuth)
15   return (
16     <div className={styles.body}>
17       <section className={styles["main-content"]}>
18         <div className={styles["main-article"]>
19           <h1>Best html,css,JavaScript courses</h1>
20           <p>Join now and get skilled</p>
21           {isAuth ? <Link to='Course/html/introduction' className={styles["main-btn"]}>Start courses</Link> :
22           <a className={styles["main-btn"]} title="потрібно зареєструватися">Start courses</a> }
23         </div>
24       </section>
25       <div className={styles.homeBox}>
26         <div className={styles.content}>
27           <div className={styles.title}>
28             <div className={styles.course}>
29               <p>In this course, you'll learn what HTML, CSS, and JavaScript are and how they interact to bring content to your browser.</p>
30             </div>
31             
32           </div>
33         </div>
34       </div>
35     </div>
36   )
37 }
38

```

Рисунок 3.7 – Код головної сторінки

Наведений код відповідає за створення головної сторінки сайту. Він включає імпорт необхідних модулів та компонентів, таких як «Header», «Link», «DropDownMenu», а також використання функцій і змінних з бібліотеки React та Redux. В середині компонента відбувається рендеринг різних елементів, таких як заголовки, текстові блоки, кнопки, зображення тощо. Також проводиться перевірка наявності аутентифікації користувача та відповідно до цього відображається відповідний текст або посилання.

Також, як приклад, можна навести код Header, зображено на рисунку 3.8.

```

1  import React from 'react';
2  import styles from './Header.module.css';
3  import { Link } from 'react-router-dom';
4  import CustomLink from '../CustomLink';
5  import DropDownMenu from '../DropDownMenu/DropDownMenu';
6  import { navItems } from '../NavItems/NavItems';
7  import { useState } from 'react';
8  import { useDispatch, useSelector } from 'react-redux';
9  import { checkAuth, logout } from '../../redux/features/auth/authSlice';
10 import { toast } from 'react-toastify';
11
12 export default function Header() {
13   const [dropDown, setDropDown] = useState(false);
14   const isAuth = useSelector(checkAuth)
15   const dispatch = useDispatch()
16
17   const logoutHandler = () => {
18     dispatch(logout())
19     window.localStorage.removeItem('token')
20     toast('Ви вийшли з системи.')
21   }
22
23   return (
24     <>
25     <div className={styles.header}>
26       <Link to="/"></Link>
27       <ul className={styles.menu}>
28         <li className={styles['header-li']} onMouseEnter={()=>{setDropDown(!dropDown)}} onMouseLeave={()=>setDropDown(false)}>Courses
29           {dropDown ? <DropDownMenu obj={navItems}/> : null}</li>
30         <CustomLink to="/"><li className={styles['header-li']}>Home</li></CustomLink>
31
32
33
34         {isAuth ? (
35           <>
36             <CustomLink to="/Post"><li className={styles['header-li']}>add Post</li></CustomLink>
37             <CustomLink to="/Blog"><li className={styles['header-li']}>Blog</li></CustomLink>
38             <CustomLink to="/posts/me"><li className={styles['header-li']}>my posts</li></CustomLink>
39             <Link to="/" style={{textDecoration: 'none'}}><li onClick={logoutHandler} className={styles['header-li']}>Вийти</li></Link>
40           </>
41         ) : (
42           <>
43             <CustomLink to="/Login"><li className={styles['header-li']}>Login</li></CustomLink>
44             <CustomLink to="/Signup"><li className={styles['header-li']}>Signup</li></CustomLink>
45           </>
46         )
47       </ul>
48     </div>
49     </>
50   )
51 }
52
53
54

```

Рисунок 3.8 – Код Header

У тілі компонента відбувається рендеринг різних елементів, таких як логотип, список з пунктами меню, кнопки входу/виходу. В компоненті також використовуються стан «dropDown», який відповідає за відображення розкривного меню при наведенні на пункт «Courses». Залежно від статусу автентифікації користувача, відображаються відповідні пункти меню, такі як «add Post», «Blog», «my posts» або «Login», «Signup».

Також в компоненті присутній обробник події для кнопки «Вийти2» («logoutHandler»), який виконує вихід з системи, видаляє збережений токен з локального сховища та виводить повідомлення за допомогою бібліотеки «react-toastify».

Отже, код створює верхню частину сайту, яка містить логотип, меню навігації та кнопки входу/виходу, і забезпечує функціональність для взаємодії з ними.

Серверна частина або back-end, у контексті React, відповідає за обробку запитів, збереження та отримання даних з бази даних, аутентифікацію користувачів та інші серверні операції, необхідні для роботи додатка. Для реалізації серверної частини використовуються інші технології та інструменти. У своєму проєкті я використовую Express.

Нижче наведений код буде демонструвати використання Express в моєму проєкті. Рисунок 3.9, демонструє модулі, які будуть використані.

```
1 import express from "express";
2 import cors from 'cors';
3 import authRouter from './routes/auth.js'
4 import postRouter from './routes/post.js'
5 import commentRouter from './routes/comment.js'
6 import fileUpload from "express-fileupload";
```

Рисунок 3.9 – Імпорт необхідних модулів

Цей розділ імпортує необхідні модулі, такі як express, cors, express-fileupload, а також маршрутизатори authRouter, postRouter і commentRouter, які визначені в окремих файлах.

На рисунку 3.10 зображено створення і конфігурація сервера.

```
8  const app = express();
9
10 app.use(express.json());
11 app.use(cors());
12 app.use(fileUpload())
13 app.use(express.static('upload'))
```

Рисунок 3.10 – Створення і конфігурація сервера

Тут створюється екземпляр сервера Express та налаштування його. Метод `app.use()` використовується для додавання проміжного ПЗ (middleware). У даному випадку, використовується `express.json()` для обробки запитів JSON, `cors()` для управління політикою міжсайтових запитів (CORS), `fileUpload()` для обробки завантаження файлів та `express.static()` для надання статичних файлів з папки «upload».

Визначення маршрутів зображено на рисунку 3.11.

```
15 app.use('/api', authRouter)
16 app.use('/api', postRouter)
17 app.use('/api', commentRouter)
```

Рисунок 3.11 – Визначення маршрутів

Тут використовується маршрутизатори `authRouter`, `postRouter` і `commentRouter`, які були імпортовані раніше, і прив'язуємо їх до шляху «/api». Це означає, що всі запити, які починаються з «/api», будуть оброблятися цими маршрутами.

Експорт серверного додатка відбувається на рисунку 3.12.

```
20 export default app;
```

Рисунок 3.12 – Експорт серверного додатку

Остаточно, експортується створений серверний додаток як модуль за допомогою синтаксису `export default`, що дозволяє використовувати його в інших частинах додатка або при запуску сервера.

Middleware – це проміжний шар програмного забезпечення, який діє як посередник між вхідними запитами та обробниками запитів. Він знаходиться між прийомом запиту сервером і виконанням обробки цього запиту. Застосування `middleware` у моєму проєкті зображено на рис. 3.13.

```
server > middlewares > JS checkAuth.js > checkAuth
1  import jwt from 'jsonwebtoken';
2
3  export const checkAuth = (req, res, next) => {
4    const token = (req.headers.authorization || '').replace(/Bearer\s?/, '')
5
6    if(token){
7      try{
8        const result = jwt.verify(token, process.env.JWT_SECRET)
9
10       req.userId = result.id
11
12       next()
13     }catch(err){
14       return res.json({message: 'Немає доступу.'})
15     }
16   }else{
17     return res.json({
18       message: 'Немає доступу.'
19     })
20   }
21 }
```

Рисунок 3.13 – Застосування `middleware`

Наведений код виконує перевірку автентифікації користувача з використанням токенів JWT (JSON Web Tokens). В функції `checkAuth`, спочатку отримується токен з заголовків запиту і виконується його перевірка. Якщо токен існує, він розшифровується за допомогою `jwt.verify` з використанням секретного ключа `process.env.JWT_SECRET`. Результат розшифрування містить ідентифікатор користувача, який потім додається до об'єкта `req` для подальшого використання.

Якщо перевірка токена успішна, виконується наступна функція `next()`, що передає управління наступному обробнику в ланцюжку маршрутизації. У випадку помилки перевірки токена або якщо токен не існує, повертається відповідь JSON з повідомленням про відсутність доступу.

Отже, загальний опис коду полягає у створенні сервера Express з налаштуваннями `middleware`, визначенні маршрутів та експорту створеного серверного додатка.

3.4 База даних

База даних – це сукупність інформації, яка організована таким чином, що до неї можна легко отримати доступ, керувати та оновлювати. Комп’ютерні бази даних зазвичай містять сукупності записів даних або файлів, що містять інформацію про операції продажу або взаємодії з конкретними клієнтами.

Головною перевагою БД є швидкість внесення та використання потрібної інформації. Завдяки спеціальним алгоритмам, які використовуються для баз даних, можна легко знаходити необхідні дані всього за декілька секунд. Також в базі даних існує певний взаємозв’язок інформації: зміна в одному рядку може спричинити зміни в інших рядках — це допомагає працювати з інформацією простіше і швидше [9].

Існують різні типи баз даних.

Ієрархічна база даних – виконує порядок ранжування або взаємозв’язок батьків і дітей для структурування даних.

Мережева база даних – схожа на ієрархічну базу даних, але з деякими змінами. Мережева база даних дозволяє дочірньому запису зв’язуватися з різними батьківськими записами, отже, дозволяючи двонаправлені відносини.

Об’єктно-орієнтована база даних – інформація зберігається об’єктно-подібним чином.

Реляційна база даних – орієнтована на таблиці, де кожен біт даних має зв'язок з кожним іншим бітом даних.

Нереляційна база даних або база даних NoSQL. Жодна база даних SQL не використовує різноманітні формати, такі як документи, графіки тощо, що забезпечує більшу гнучкість і масштабованість для проектування бази даних.

Однак бази даних широко поділяються на два основні типи, а саме: реляційні бази даних та нереляційні бази даних.

Реляційні бази даних – це БД, що зберігають дані у таблиці і рядках, які також називаються записами, наприклад, PostgreSQL, SQLite, MySQL та MariaDB.

Нереляційні бази даних відрізняються від реляційних баз даних тим, що вони не використовують таблиці, рядки, первинні або зовнішні ключі. Замість цього, нереляційна база даних оперує моделлю сховища, яка оптимізована для задоволення конкретних вимог до типу даних, які зберігаються.

Для збереження даних користувачів, я обрала нереляційну базу даних MongoDB.

MongoDB – це документо-орієнтована база даних, яка зберігає дані у вигляді документів в форматі JSON (JavaScript Object Notation). Вона дозволяє зберігати інформацію у вигляді складних об'єктів, які можуть містити різноманітні поля, що дозволяє зберігати дані у більш структурованому вигляді порівняно з традиційними реляційними базами даних.

MongoDB має багато переваг перед реляційними базами даних, зокрема:

- висока швидкість роботи: MongoDB працює швидше, ніж більшість реляційних баз даних, оскільки документи можуть бути збережені в одному дисковому блоку;
- простота використання: MongoDB дуже легко встановлювати та використовувати. Крім того, вона має приємний інтерфейс для роботи з даними;

- гнучкість: MongoDB дозволяє зберігати будь-яку кількість полів у документах, що робить його більш гнучким ніж традиційні реляційні бази даних.

Існують також інші документо-орієнтовані бази даних, які можуть бути альтернативами MongoDB. Декілька з них:

- Couchbase: є документо-орієнтованою базою даних з високою продуктивністю, що працює у розподіленому середовищі;
- Apache Cassandra: є ключ-значенням базою даних з високою доступністю, яка дозволяє зберігати великі обсяги даних у розподілених середовищах;
- Amazon DynamoDB: є повністю керованою базою даних з розміщенням у реальному часі та відмінною масштабованістю.

Я вважаю, що MongoDB є доречним варіантом бази даних для мого проєкту з таких причин: вона має гнучку структуру даних, що дозволяє зручно працювати з різними типами і форматами інформації; вона також відзначається простотою використання, що спрощує процес розробки та підтримки; крім того, MongoDB демонструє високу продуктивність, що забезпечує швидкий доступ до даних; і, нарешті, вона масштабована, що означає, що вона може легко розширюватись і впоратись зі зростаючим обсягом даних.

3.5 Реалізація бази даних

MongoDB використовує схему бази даних, яку називають «гнучкою схемою». MongoDB не має фіксованих таблиць і зміни схеми можуть бути виконані безпосередньо в документах бази даних. Замість табличної структури MongoDB використовує колекції документів, де кожен документ представляє сутність з набором ключ-значення полів. Схема бази даних зображена на рисунку 3.14.

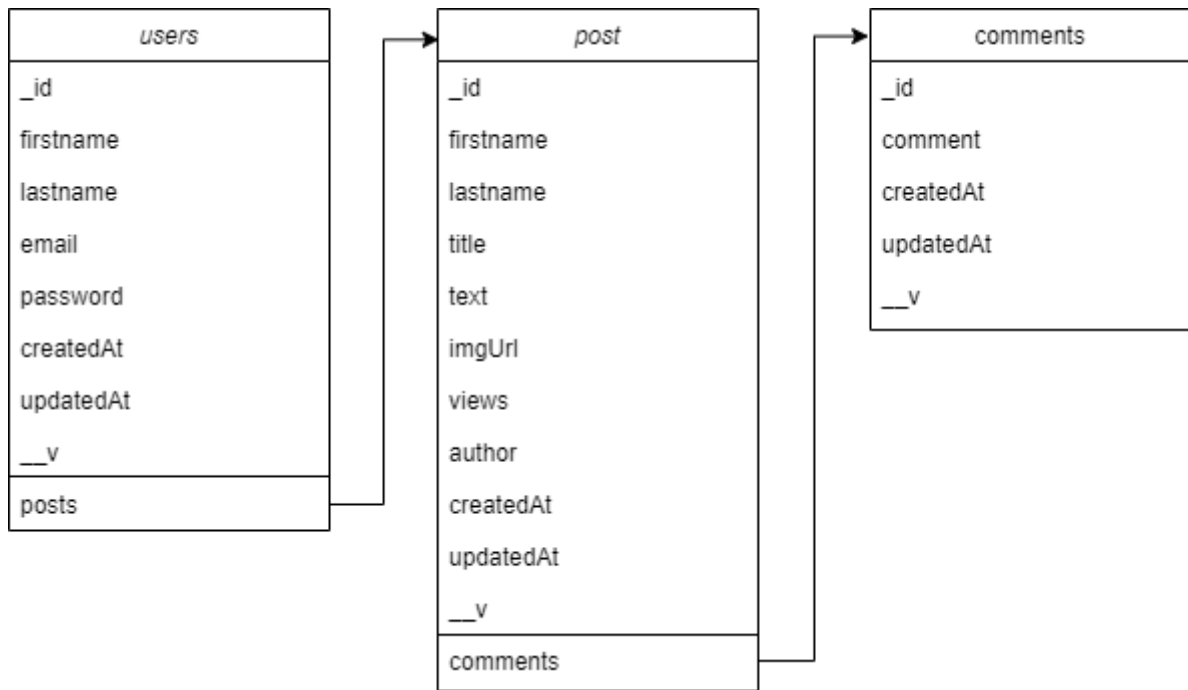


Рисунок 3.14 – Схема бази даних

Для підключення до БД використовую mongoose. Mongoose – це об’єктно-реляційна бібліотека для роботи з MongoDB у JavaScript-додатках. Вона дозволяє встановлювати з’єднання з MongoDB, визначати моделі даних і виконувати різноманітні операції з базою даних, такі як додавання, оновлення, видалення та запити. Mongoose надає зручний спосіб взаємодії з MongoDB, допомагаючи спростити розробку і керування даними в програмах на JavaScript.

На рис. 3.15 продемонстровано підключення до бази даних.

```

async function start(){
  try{
    await mongoose.connect(`mongodb+srv://${DB_USER}:${DB_PASSWORD}@cluster0.gsetepn.mongodb.net/${DB_NAME}?retryWrites=true&w=majority`)
  }
  server.listen(PORT, ()=>{
    console.log(`Server started on port ${PORT}`)
  })
}
  
```

Рисунок 3.15 – Підключення до бази даних

Наприклад, для створення коментарю я використовую код, який зображено на рисунку 3.16.

```

1  import Comment from "../models/Comment.js";
2  import Post from "../models/Post.js";
3
4  export const createComment = async (req, res) => {
5      try {
6          const {postId, comment} = req.body
7
8          if(!comment){
9              res.json({message: 'Коментар не може бути порожнім.'})
10         }
11         const newComment = new Comment({comment})
12         await newComment.save()
13
14         try {
15             await Post.findByIdAndUpdate(postId, {
16                 $push: {comments: newComment._id}
17             })
18         } catch (error) {
19             console.log(error)
20         }
21
22         res.json(newComment)
23     } catch (error) {
24         res.json({message: 'Щось пішло не так.'})
25     }
26 }

```

Рисунок 3.16 – Додавання коментаря

Отже, MongoDB використовує гнучку схему бази даних, що дозволяє змінювати схему безпосередньо в документах бази даних. Замість таблиць, MongoDB використовує колекції документів, де кожен документ має набір ключ-значення полів. Для зручної роботи з MongoDB у JavaScript-додатках можна використовувати бібліотеку Mongoose, яка дозволяє підключатися до бази даних, визначати моделі даних і виконувати різноманітні операції з даними. Підключення до бази даних можна здійснити за допомогою методів, які надає Mongoose. Використання Mongoose допомагає спростити розробку і керування даними в програмах на JavaScript.

4 ПРИКЛАД РОБОТИ ДОДАТКУ

Нижче представлені знімки екрану веб-додатка, який працює у демонстраційному режимі.

Перший екран, який зустрине користувач, є головною сторінкою, де він матиме можливість зареєструватися або увійти до свого акаунту, оскільки доступ до курсів обмежений для незареєстрованих користувачів (рис. 4.1).



Рисунок 4.1 – Головний екран сайту

На рисунку 4.2 зображена форма Логіну, яка надає користувачам можливість ввести свої облікові дані для отримання доступу до системи.

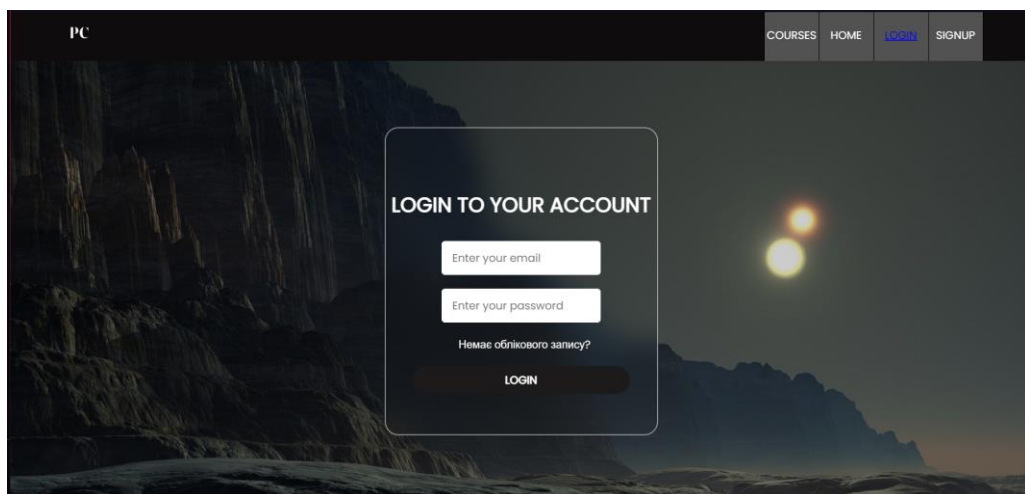


Рисунок 4.2 – Форма логіну

На рисунку 4.3 показано можливість зареєструватися для користувачів, які ще не мають облікового запису. Цей варіант надає їм можливість створити новий аккаунт і отримати повний доступ до всіх функцій системи. Для переходу до цієї форми, користувач може натиснути на посилання «Немає облікового запису?» або перейти безпосередньо на сторінку реєстрації (Signup).

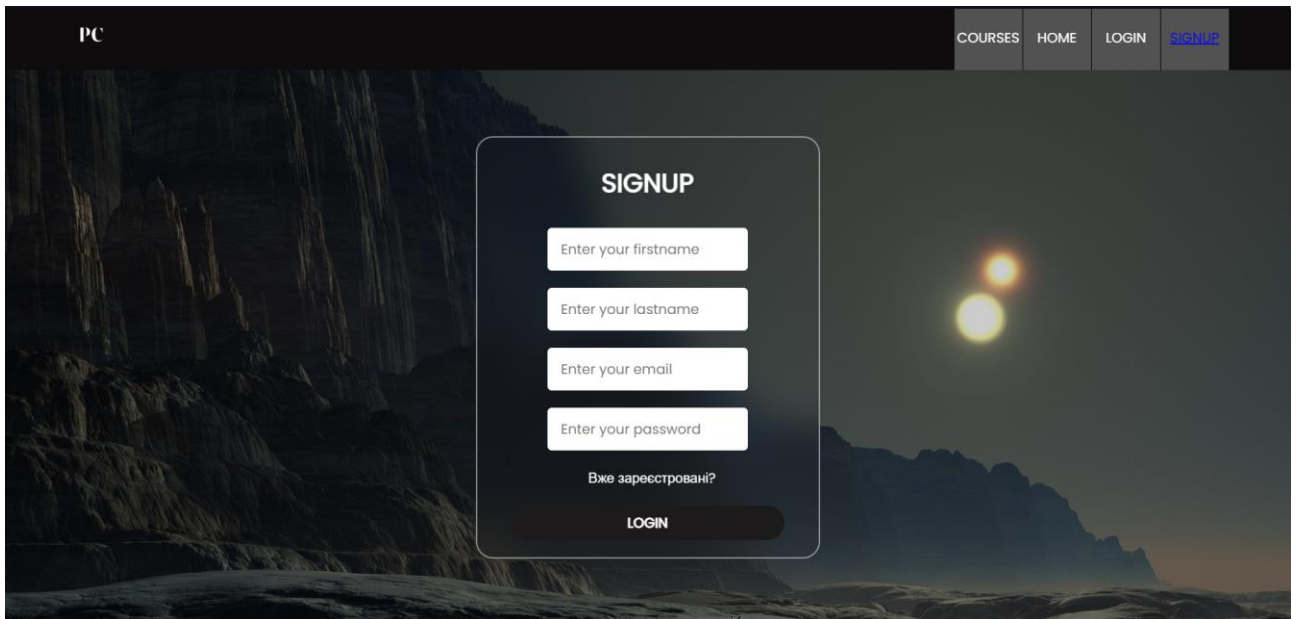


Рисунок 4.3 – Форма реєстрації

Після успішного входу в особистий аккаунт, меню веб-сайту змінюється, додавши нові пункти, які стають доступними для зареєстрованих користувачів. На рисунку 4.4 зображено оновлене меню, яке надає розширені функції та ресурси. Зареєстровані користувачі можуть звертатися до цих нових пунктів для отримання інформації про свої пости, персональні повідомлення та доступні курси та матеріали. Це оновлене меню забезпечує зручну навігацію та розширені можливості для користувачів після входу в їх особистий аккаунт.



Рисунок 4.4 – Нові пункти меню

При наведенні курсору на пункт «Courses» в меню, з'являється випадаюче меню, яке дозволяє користувачеві обрати між доступними курсами та лабораторними роботами, яке зображено на рисунку 4.5. Це надає можливість вибору конкретного курсу або лабораторної роботи залежно від інтересів та потреб користувача.



Рисунок 4.5 – Випадаюче меню курсів

Після вибору користувачем потрібного курсу, відкривається сторінка з курсами, де ліворуч розташовані теми, а праворуч міститься відповідна інформація до кожної з цих тем, рисунок 4.6. Ця організація сторінки дозволяє користувачеві легко навігуватися по різних темах курсу та отримувати необхідну інформацію. За допомогою лівої панелі з темами, користувачі можуть швидко переходити між різними частинами курсу, а праворуч вони знаходять детальну інформацію, матеріали та завдання, що стосуються обраної теми. Ця структура сторінки забезпечує зручний доступ до вмісту курсу та полегшує процес навчання користувачів.

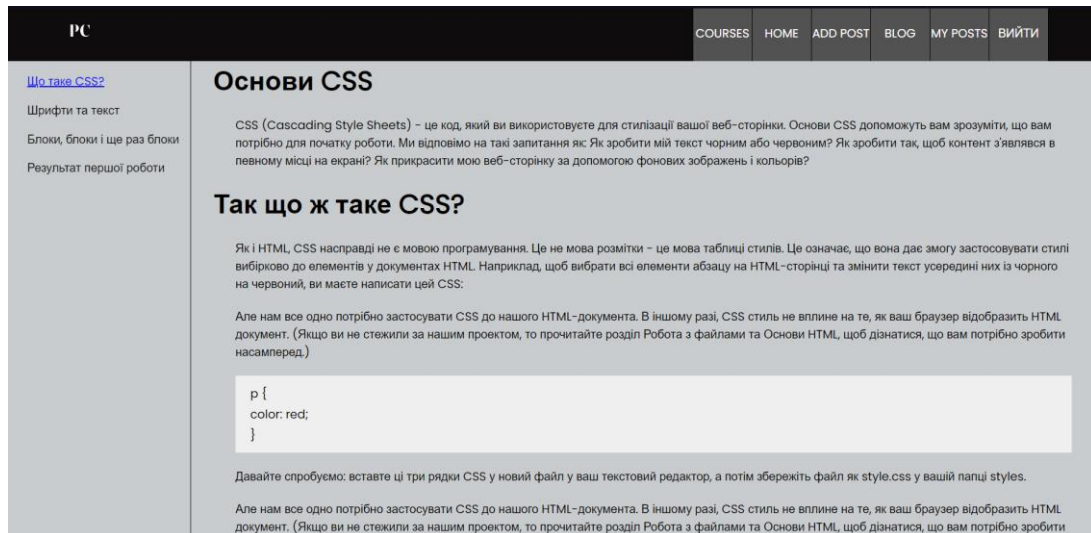


Рисунок 4.6 – Організація сторінки курсу

При виборі опції «Labs» замість курсів, користувач переходить на сторінку лабораторних робіт, де ліворуч відображається перелік доступних лабораторних робіт, а праворуч, після вибору конкретної лабораторної роботи, відкривається сама лабораторна робота з завданнями та посиланням на GitHub з вихідним кодом (рис. 4.7). Ця структура сторінки забезпечує зручний доступ до лабораторних робіт, дозволяючи користувачам швидко вибрати потрібну лабораторну роботу та отримати повну інформацію про неї. Посилання на GitHub з вихідним кодом дозволяє зручно взаємодіяти з кодом та виконувати завдання на основі цієї лабораторної роботи.

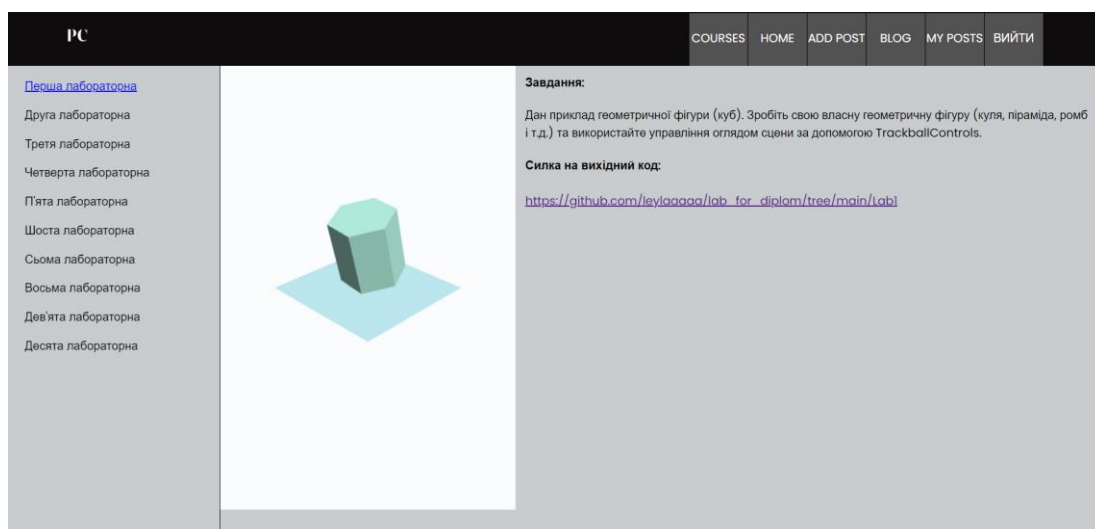


Рисунок 4.7 – Організація сторінки курсу

В проєкті також включена можливість додавати пости, як показано на рисунку 4.8. Користувачі мають змогу створювати та публікувати свої власні пости, де вони можуть ділитися інформацією, дослідженнями, думками та іншими відповідними матеріалами. Додавання постів створює можливості для взаємодії та обміну знаннями між користувачами проєкту. Зображений на рисунку 4.8 інтерфейс дозволяє легко створювати та публікувати пости, а також додавати в них зображення для наочності та додаткової інформації.

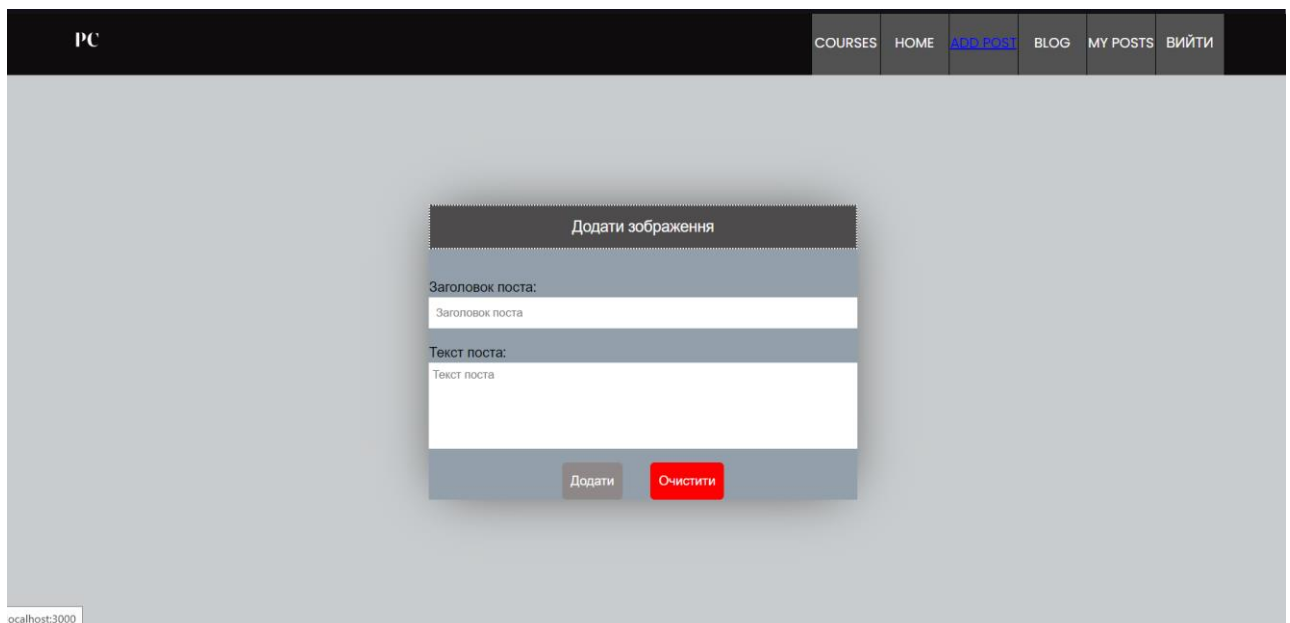


Рисунок 4.8 – Сторінка створення постів

Для перегляду наявних постів необхідно перейти до вкладки «Blog», яка зображена на рисунку 4.9. Ця вкладка надає зручний доступ до колекції постів, де користувачі можуть ознайомитися зі статтями, думками та матеріалами, що були опубліковані іншими учасниками. Рисунок 4.9 демонструє інтерфейс вкладки «Blog», де можна переглядати список постів за датою публікації або за іншими параметрами, що допомагають зорієнтуватися і знайти цікавий контент.

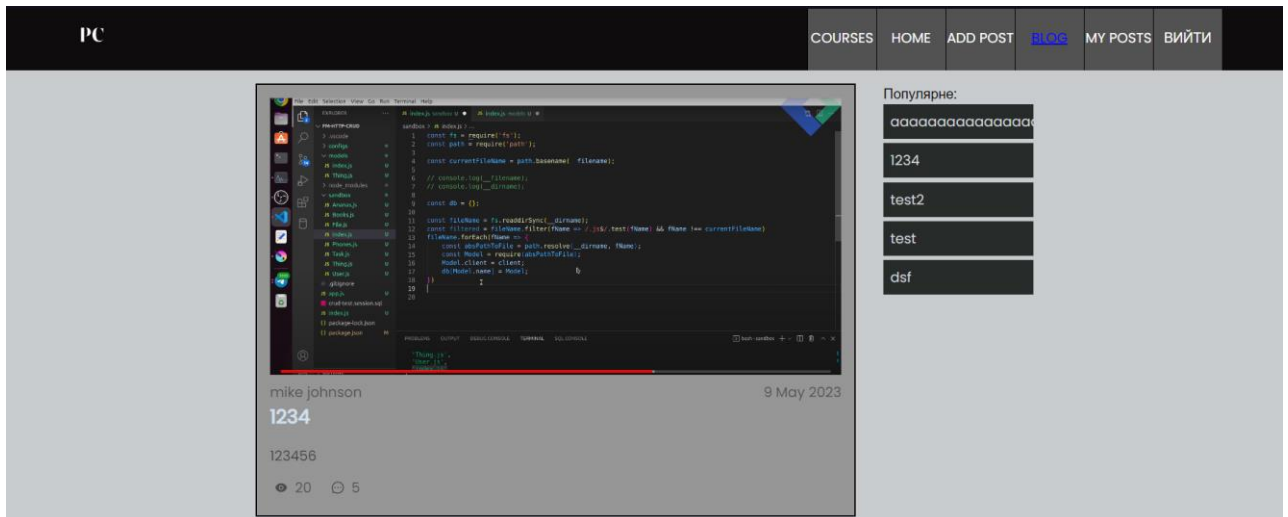


Рисунок 4.9 – Сторінка перегляду блогу

Після вибору посту зі вкладки «Blog», користувач може переглянути завантажене зображення автором, тему посту, опис і також залишити коментар до цього посту, рисунок 4.10. Крім текстового контенту, автори постів мають можливість включати ілюстрації або фотографії, які доповнюють їхній пост. Користувачі можуть ознайомитися з інформацією про пост, його темою та описом, а також матимуть змогу висловити свої думки та залишити коментар під цим постом. Це створює можливість для взаємодії, обговорення та обміну ідеями між користувачами.

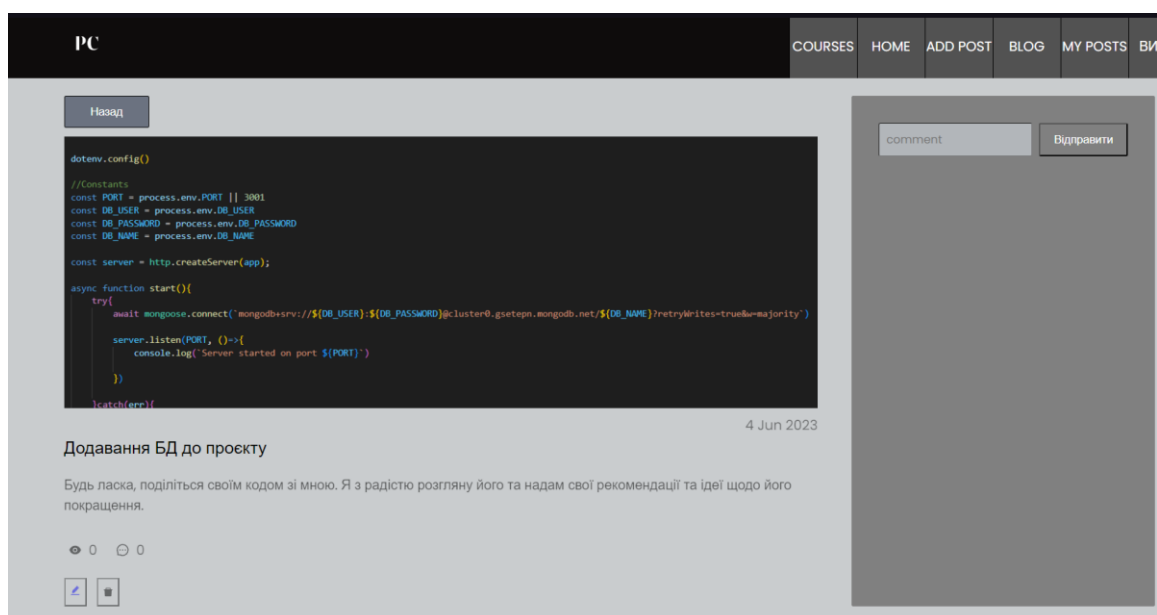


Рисунок 4.10 – Сторінка перегляду посту

Для користувачів, які створюють пости, доступна можливість перегляду всіх їх створених постів. Цю функціональність можна знайти на рисунку 4.11, де зображено сторінку зі списком всіх постів, які були створені користувачем. Це дозволяє користувачеві швидко переглянути і керувати своїми власними постами, забезпечуючи зручний спосіб організації та навігації через власний КОНТЕНТ.

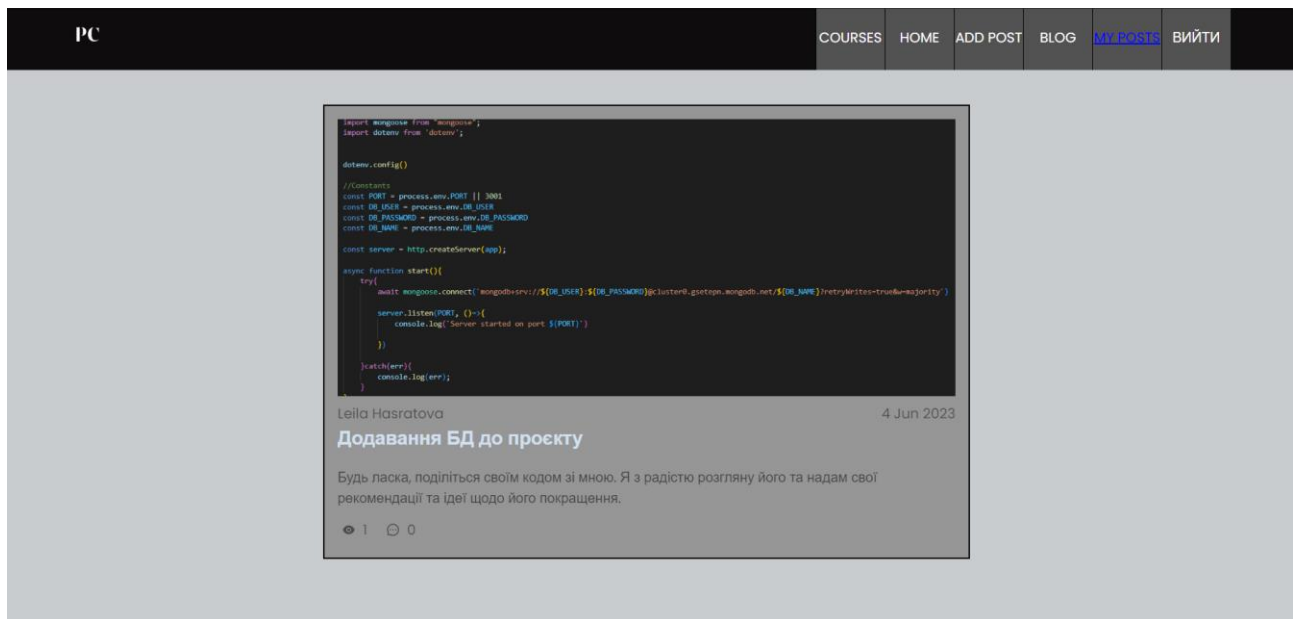


Рисунок 4.11 – Сторінка перегляду посту

Отже, у даному розділі проєкту було розглянуто різноманітні функції, що дозволяють користувачам використовувати платформу ефективно. Вони включають можливість реєструватися в системі, входити до свого існуючого акаунту, переглядати доступні курси та лабораторні роботи, а також читати та створювати пости. Цей функціонал надає користувачам широкі можливості для отримання інформації, взаємодії та спілкування на платформі. Всі ці функції допомагають підтримувати активну спільноту користувачів та створюють зручне середовище для навчання, обміну думками та розвитку знань. Завдяки цим функціям, користувачі можуть максимально використовувати потенціал проєкту і здобувати нові знання та навички.

ВИСНОВКИ

Отже, розробка онлайн курсу «Комп'ютерна графіка в WEB» є дуже важливою задачею в сучасному світі. З розвитком інтернету та електронної комунікації, веб-розробка та веб-дизайн стали надзвичайно важливими, а вимоги до веб-дизайнерів та розробників зростають з кожним днем.

Онлайн курси з комп'ютерної графіки веб-дизайну надають можливість людям отримати якісну освіту в цій галузі в будь-який зручний для них час та місці, а також від досвідчених фахівців з усього світу.

Розробка онлайн курсу «Комп'ютерна графіка в WEB» потребує багато ресурсів, таких як час, гроші, знання та досвід. Проте, розробка такого курсу допоможе підготувати нову хвилю веб-дизайнерів та розробників, які зможуть створювати якісні веб-сайти та додатки, що відповідають сучасним стандартам. Він може стати цінним ресурсом для всіх, хто бажає розвинути свої навички у сфері веб-дизайну та розробки. Цей курс може охоплювати різноманітні аспекти комп'ютерної графіки, які стосуються веб-середовища.

У процесі дослідження були використані різні методи, такі як аналіз предметної області, вивчення та узагальнення і моделювання. Основними засобами реалізації веб-додатку стали CSS, JavaScript, HTML, React та MongoDB, які дозволили створити функціональний та інтуїтивно зрозумілий інтерфейс для користувачів.

У результаті роботи була розроблена база даних, а також реалізовано основні функції та можливості веб-додатку для проведення онлайн курсу з комп'ютерної графіки. Це включає в себе перегляд курсів, доступ до матеріалів, можливість додавання та коментування постів, а також управління користувачами.

Кваліфікаційна робота пропонує цінний внесок у сферу онлайн-навчання та надає підґрунтя для подальшого розвитку та вдосконалення веб-додатку для курсу «Комп'ютерна графіка в Web».

ПЕРЕЛІК ПОСИЛАНЬ

1. What Is Coursera? URL: <https://www.coursera.org/articles/what-is-coursera> (дата звернення: 23.03.2023).
2. THE 10 BEST ONLINE WEB DESIGN CLASSES OF 2023 URL: <https://www.intelligent.com/best-online-courses/web-design-classes/> (дата звернення: 27.03.2023).
3. Haverbeke M. Eloquent JavaScript, 3rd Edition: : A Modern Introduction to Programming, 2018. 472 p.
4. React. URL: <https://react.dev> (дата звернення: 28.03.2023).
5. What is Angular? URL: <https://angular.io/guide/what-is-angular> (дата звернення: 29.03.2023).
6. Діаграма прецедентів. URL: https://www.wikiwand.com/uk/Діаграма_прецедентів (дата звернення: 30.03.2023).
7. Best Practices for Minimalist Website Design. URL: <https://medium.muz.li/best-practices-for-minimalist-website-design-9e8ea07e17c2?gi=1bf93b8c1c71#:~:text=Minimalism%20in%20web%20design%20means,words%2C%20do%20more%20with%20less> (дата звернення: 10.04.2023).
8. Мухін В. В. Програмування комп'ютерної графіки у WEB. URL: <https://moodle.znu.edu.ua/course/view.php?id=22> (дата звернення: 15.04.2023).
9. Герасимик І. В. Що таке база даних? URL: <http://apeps.kpi.ua/shco-take-basa-danykh> (дата звернення: 01.05.2023).