

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

КВАЛІФІКАЦІЙНА РОБОТА ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

РОЗРОБКА ВЕБ-МАГАЗИНУ
З ПРОДАЖУ КУРСІВ ПРОГРАМУВАННЯ

DEVELOPMENT OF A WEB STORE
FROM THE SALE PROGRAMMING COURSES

спеціальність 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

освітня програма «Інженерія програмного забезпечення»
(назва освітньої програми)

Виконав: здобувач вищої освіти
Групи ПІЗ-41
Устимець Дем'ян Богданович


(підпис)

Керівник
д.т.н., професор
Гордєєв Олександр Олександрович

(підпис)

Кваліфікаційну роботу
допущено до захисту
«8» 06 2023 р.
Гарант освітньої програми:
к.т.н., доцент
Ліщина Наталія Миколаївна

(підпис)

Луцьк – 2023 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 121 Інженерія програмного забезпечення

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

«2» 02 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Устимцю Дем'яну Богдановичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: «Розробка веб-магазину з продажу курсів програмування»

Керівник роботи: д.т.н., професор Гордєєв Олександр Олександрович

затверджені наказом закладу вищої освіти від «23» грудня 2022р. № 961-01-02

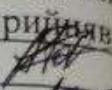
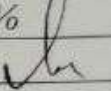
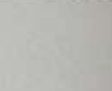
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи «08» червня 2023 р.

3. Вихідні дані до роботи: вимоги до розробки програмного забезпечення, вимоги до функціонування програмного засобу. Технології розробки клієнтської частини – HTML, TAILWIND, POSTCSS, JS, Node.js.

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

Аналіз сучасного стану проблеми, існуючих методів і засобів її розв'язання, аналіз і вибір засобів розробки веб-сайтів, опис функціонального наповнення об'єкта, розробка й обґрунтування системного наповнення, функціонально-структурна схема роботи веб-сайту, розробка дизайну веб-сайту.

5. Перелік графічного матеріалу: 29 рис., 2 табл., 1 дод., 14 джерел.

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		Завдання видав	Завдання прийняв
Аналіз предметної області	Гордєєв О.О.		
Специфікація вимог до розробленої системи	Гордєєв О.О.		
Розробка об'єкта проектування	Гордєєв О.О.		
Нормоконтроль	Гордєєв О.О.		
Гарант ОП	Ліщина Н.М.		
Показник запозичень тексту		23.7%	
Академічна доброчесність	Яцук А.А.		

7. Дата видачі завдання « 2 » лютого 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної роботи	Строк виконання етапів роботи	Примітка
1.	Провести огляд літературних джерел по темі кваліфікаційної роботи бакалавра	05.02.2023 р.	Виконано
2.	Провести порівняльну характеристику програмних продуктів	27.02.2023 р.	Виконано
3.	Розробити функціональну схему роботи об'єкта проектування	10.03.2023 р.	Виконано
4.	Описати програмне та апаратне середовище функціонування об'єкта проектування	17.04.2023 р.	Виконано
5.	Практична реалізація об'єкта проектування	18.05.2023 р.	Виконано
6.	Тестування та налагодження об'єкта розробки	01.06.2023 р.	Виконано
7.	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедрі	08.06.2023 р.	Виконано

Здобувач вищої освіти:


(підпис)

(Устимець Д.Б.)
(прізвище та ініціали)

Керівник кваліфікаційної роботи:


(підпис)

(Гордєєв О.О.)
(прізвище та ініціали)

Міністерство освіти і науки України
Луцький національний технічний університет

Рецензія
на кваліфікаційну роботу

Здобувачвищоїосвіти:

Устимець Дем'ян Богданович

Тема: Розробка веб-магазину з продажу курсів програмування / Development of a web store from the programming courses.

Коротка характеристика кваліфікаційної роботи:

Робота складається з трьох розділів, протягом яких дипломантом здійснюється огляд поставленої перед ним проблеми, пошук шляхів її вирішення та опис їх реалізації. Для веб-застосунку було використано наступні програмні засоби: HTML для структури та маркування веб-сторінок, TAILWIND для швидкого стилізування елементів з використанням класів, POSTCSS для оптимізації та трансформації CSS-коду, JavaScript (JS) для реалізації основної логіки застосунку, а також NODE для створення серверної частини веб-застосунку.

Самостійні розробки і пропозиції автора:

Кандидат пропонує використовувати JS для логіки інтернет-магазину курсів програмування з динамічним кошиком, фільтрацією курсів та оновленнями в режимі реального часу. Для серверної частини рекомендується використовувати NODE.js для безперешкодної комунікації та функціоналу аутентифікації, безпечної обробки платежів та керування базами даних. За допомогою HTML, TAILWIND, POSTCSS та JS кандидат прагне створити функціональний та привабливий інтернет-магазин курсів програмування.

Практичне значення роботи:

Кандидатом було створено веб-сервіс для продажу курсів програмування з каталогом курсів, кошиком та замовленнями, що допомагає людям розвивати свої навички в програмуванні.

Недоліки:

У роботі не виявлено новачі – вона більш походить на альтернативу існуючим варіантам вирішення проблеми, проте слід помітити що складна структура роботи добре висвітлює вміння дипломанта.

Загальний висновок:

Рівень якості виконаної роботи є задовільним. Усі поставлені завдання успішно й повноцінно виконані. Кваліфікаційна робота виконана на якісному рівні тож рекомендується до захисту та заслуговує на високу оцінку.

Рецензент: _____ доцент, к.т.н. Поліщук М.М.
(прізвище, ім'я, по-батькові, посада)

Рецензент кваліфікаційної роботи


(підпис)

«08» червня 2023 р.

АНОТАЦІЯ

Устимець Д.Б. Розробка веб-магазину з продажу курсів програмування.
Рукопис.

Кваліфікаційна робота бакалавра. Кваліфікаційна робота за спеціальністю 121 Інженерія програмного забезпечення. Луцьк 2023. 64 с.

Кваліфікаційна робота присвячена розробці та реалізації веб-магазину з продажу курсів програмування з використанням сучасних технологій. У роботі розглядаються проблематика веб-додатків та завдання, пов'язані з їх розв'язанням. Описано процес розробки веб-магазину та досліджено технології, які використовуються для його реалізації, зокрема JavaScript та Node. Детально розкрито сутність використання цих фреймворків для мови програмування JavaScript. Також проведено аналіз та вивчення суб'єкту управління базами даних.

Ключові слова: веб-магазин, курси програмування, база даних, HTML, CSS, JavaScript, Node.

ABSTRACT

Ustymets D.B. Development of an Online Store for Selling Programming Courses. Manuscript.

Bachelor's Qualification Work. Specialization: 121 Software Engineering. Lutsk 2023. 64 p.

The qualification work is dedicated to the development and implementation of an online store for selling programming courses using modern technologies. The paper addresses the issues related to web applications and the tasks associated with their solutions. The process of developing the online store is described, and the technologies used for its implementation, particularly JavaScript and Node, are investigated. The essence of utilizing these frameworks for the JavaScript programming language is thoroughly explored. Additionally, an analysis and study of the database management subject are conducted.

Keywords: online store, programming courses, database, HTML, CSS, JavaScript, Node.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ	10
1.1 Аналіз сучасного стану проблеми	10
1.2 Постановка завдання на кваліфікаційну роботу бакалавра	11
1.3 Огляд існуючих рішень	12
Висновок до розділу 1	17
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ	19
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення.....	19
2.2 Вибір засобів, методів та алгоритмів вирішення поставленого завдання	22
2.3 Обрання мов програмування	24
2.4 Структура об'єкта.....	25
Висновки до розділу 2.....	27
РОЗДІЛ 3 РОЗРОБКА ВЕБ ДОДАТКУ	28
3.1 Практична реалізація об'єкта проектування.....	28
3.1.1 Визначення функціональних вимог	28
3.1.2 Визначення нефункціональних вимог	29
3.2 Визначення компонентів програми	29
3.2.1 Визначення модулів та їх взаємодії	30
3.2.2 Визначення архітектурного шаблону	31
3.3 Розробка та тестування	32
3.3.1 Реалізація функціональності	32
3.3.2 Тестування	33
3.4 Встановлення необхідної конфігурації	33
3.5 Опис і реалізація основних компонентів	36
3.5.1 Модуль Sign Up	36
3.5.2 Модуль LandingPage	39
3.5.3 Модуль Cart.....	44
3.5.4 Модуль AfterLogin.....	47

3.5.5 Реалізація інших компонентів.....	49
Висновки до розділу 3.....	53
ВИСНОВКИ	55
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	57
ДОДАТКИ	58

ВСТУП

Сучасний світ високих технологій вимагає постійного розвитку та набуття нових навичок, особливо в галузі програмування. Програмування стає все більш важливим і високопопитним навичкою, оскільки воно знаходить застосування в різних сферах, починаючи від розробки програмного забезпечення та веб-додатків до штучного інтелекту та інтернету речей. Щоб крокувати в ногу з часом та розкрити свій потенціал у цій сфері, студенти та професіонали потребують доступу до якісного навчання та курсів програмування.

Темою наведеної кваліфікаційної роботи є розробка веб-магазину з продажу курсів програмування, що має на меті підвищення доступності навчання програмуванню та зручності для студентів та працівників, які бажають розвивати свої навички у цій галузі.

Актуальність теми полягає в тому, що зростаюча популярність онлайн-освіти і програмування вимагає ефективних платформ для продажу та навчання.

Метою кваліфікаційної роботи бакалавра є розробка веб-магазину з продажу курсів програмування з метою підвищення доступності навчання програмуванню та зручності для студентів та працівників, які бажають розвивати свої навички у цій галузі.

Завданням данної роботи:

- розробка дизайну та структури веб-магазину з продажу курсів програмування;
- розробити функціональних можливостей веб-магазину, включаючи обробку замовлень, реєстрацію користувачів, оплату та надання доступу до курсів;
- забезпечення інтеграції з платіжними системами для онлайн-платежів;
- розробка системи управління контентом для додавання, редагування та видалення курсів та їх інформації.

Об'єктом дослідження є веб-магазин з продажу курсів програмування.

Предметом дослідження є процес розробки веб-магазину та його функціональні можливості, включаючи веб-дизайн, функціональність, інтеграцію з платіжними системами, систему управління контентом та захист даних користувачів.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

Тема кваліфікаційної роботи «Розробка веб-магазину з продажу курсів програмування» є дуже актуальною у сучасному світі, де інформаційні технології набувають все більшого значення і впливу на різні сфери життя. Програмування є однією з ключових навичок, яка вимагається в багатьох професіях, починаючи від розробки програмного забезпечення до аналізу даних та штучного інтелекту.

Зростання популярності програмування породжує зростаючу потребу в якісному навчанні та доступі до навчальних ресурсів. Однак, традиційні методи навчання, такі як походи до фізичних шкіл або університетів, можуть бути обмеженими з точки зору доступності та гнучкості. Веб-магазин з продажу курсів програмування вирішує ці обмеження, надаючи користувачам можливість навчатись в онлайн-форматі, зручно пристосовуючи навчання до свого графіку та потреб.

Подальша актуальність даної теми полягає у розширенні можливостей навчання програмування, забезпеченні доступності для широкого кола студентів та професіоналів, а також забезпеченні якості та ефективності процесу навчання. Розробка веб-магазину з продажу курсів програмування відкриває нові перспективи для навчання та розвитку в галузі програмування, допомагаючи студентам та професіоналам покращити свої навички та розвинути свій потенціал.

У зв'язку зі стрімким розвитком сфери програмування та великою кількістю людей, які зацікавлені у вивченні цієї галузі, розробка веб-магазину з продажу курсів програмування стає важливою задачею. Ця робота спрямована на розробку функціонального та ефективного веб-магазину, який забезпечить

доступність та зручність навчання програмування для широкого кола користувачів.

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

Метою даної кваліфікаційної роботи є розробка веб-магазину з продажу курсів програмування з використанням сучасних технологій. Для досягнення цієї мети необхідно вирішити наступні завдання:

- описати проблематику веб-додатків та завдання, пов'язані з розробкою веб-магазину з продажу курсів програмування;
- розробити дизайн та структуру веб-магазину, забезпечити його зручність та естетичний вигляд;
- розробити функціональні можливості веб-магазину, включаючи обробку замовлень, реєстрацію користувачів, оплату та надання доступу до курсів;
- забезпечити інтеграцію з платіжними системами для прийому онлайн-платежів та забезпечення безпеки транзакцій;
- розробити систему управління контентом для додавання, редагування та видалення курсів та інформації про них;
- забезпечити можливість оцінки та залишення відгуків користувачами про курси;
- розробити механізми авторизації та захисту особистої інформації користувачів;
- провести тестування та оптимізацію веб-магазину для забезпечення швидкості та надійності його роботи;
- дослідити та проаналізувати використання сучасних технологій у розробці веб-магазинів, зокрема JavaScript та Node;
- оцінити ефективність розробленого веб-магазину з продажу курсів програмування на основі показників його роботи та заданих критеріїв якості.

Виконання цих завдань дозволить розробити функціональний та зручний веб-магазин, який забезпечить ефективний процес продажу курсів програмування та задовольнить потреби користувачів.

1.3 Огляд існуючих рішень

На сьогоднішній день існує значна кількість рішень у галузі веб-магазинів для продажу курсів програмування. Деякі з них вже здобули велику популярність та успіх на ринку, інші ще розвиваються та пропонують нові підходи до навчання програмування. Огляд існуючих рішень є важливим етапом кваліфікаційної роботи для визначення їх переваг, недоліків та можливостей для подальшого розвитку.

Одним з найпопулярніших існуючих рішень є платформа Udemy, яка продемонстрована на рисунку 1.1. Вона надає широкий вибір курсів програмування, які розробляються викладачами з усього світу. Udemy відомий своїми демократичними цінами та великим розмаїттям тематик, що пропонуються користувачам. Однак, у нього є свої недоліки, зокрема, відсутність контролю якості навчальних матеріалів та нестабільність цін.

Розглянемо детальніше

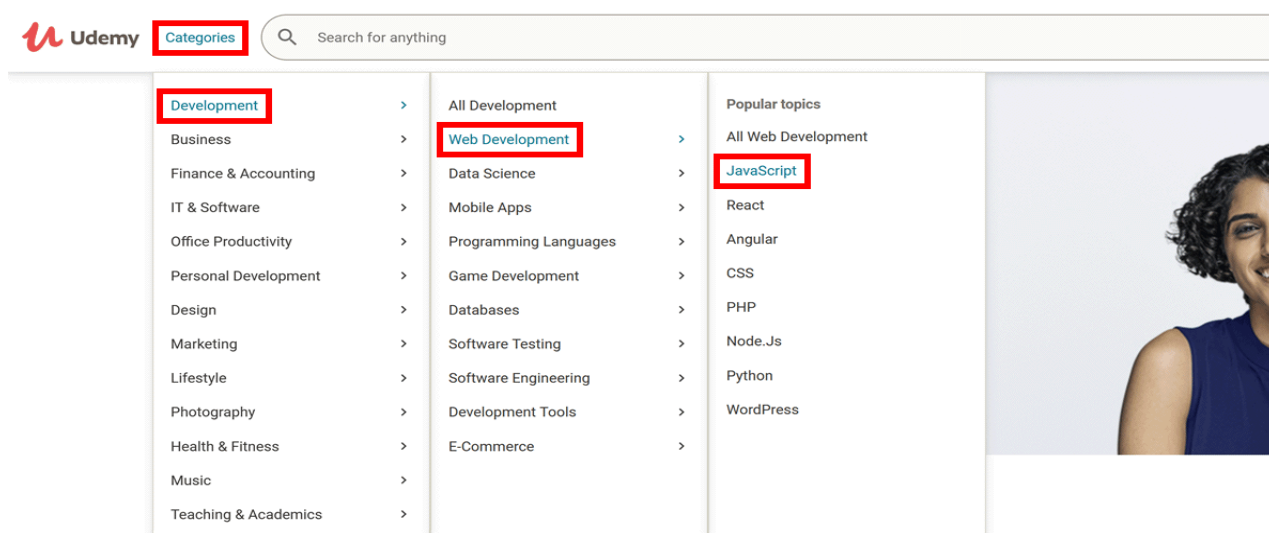


Рисунок 1.1 – Інтерфейс Udemy [1]

Переваги:

1) широкий вибір курсів. Udeemy має велику бібліотеку курсів, які наведено на рисунку 1.2 з різних галузей програмування, що дозволяє користувачам знайти підходящий курс для своїх потреб;

2) доступна цінова політика. Udeemy пропонує курси за демократичними цінами, що робить навчання доступним для широкого кола користувачів;

3) гнучкий графік. Користувачі можуть вчитися власним темпом та обирати зручний час для навчання, оскільки більшість курсів на Udeemy доступні у форматі самостійного навчання.

Courses to get you started

Most popular New Beginner Favorites



German Made Simple: A Complete Course for Serious...

Ingo Depner

4.5 ★★★★★ (4,861)

€17.99 €99.99



Perfect Your German: Tips & Tricks to Avoid Common...

Ingo Depner

4.8 ★★★★★ (1,260)

€16.99 €94.99



German for beginners (A1 - Part 1/5)

Juliane Klingenberg Nery

4.7 ★★★★★ (1,871)

€17.99 €99.99

Рисунок 1.2 – Найпопулярніші курси [1]

Недоліки:

1) відсутність контролю якості. Зміст курсів на Udeemy розробляють різні інструктори, і якість матеріалів може варіюватися від курсу до курсу;

2) нестабільність цін. Ціни на курси на Udeemy можуть змінюватися від часу до часу, що може впливати на бюджет користувача при покупці курсу;

3) відсутність академічного визнання. Курси на Udeemy не завжди мають академічне визнання, що може бути важливим для деяких студентів чи професіоналів.

Інший приклад існуючого рішення – платформа Coursera, яка зображена на рисунку 1.3. Вона співпрацює з провідними університетами та навчальними закладами, що надає користувачам доступ до високоякісних курсів, приклад

курсів наведено на рисунку 1.4, з програмування. Coursera відрізняється гнучкістю навчання, можливістю отримання сертифікатів та доступом до спеціалізацій. Однак, вартість курсів на Coursera може бути досить високою, що утруднює доступ до навчання для деяких користувачів.

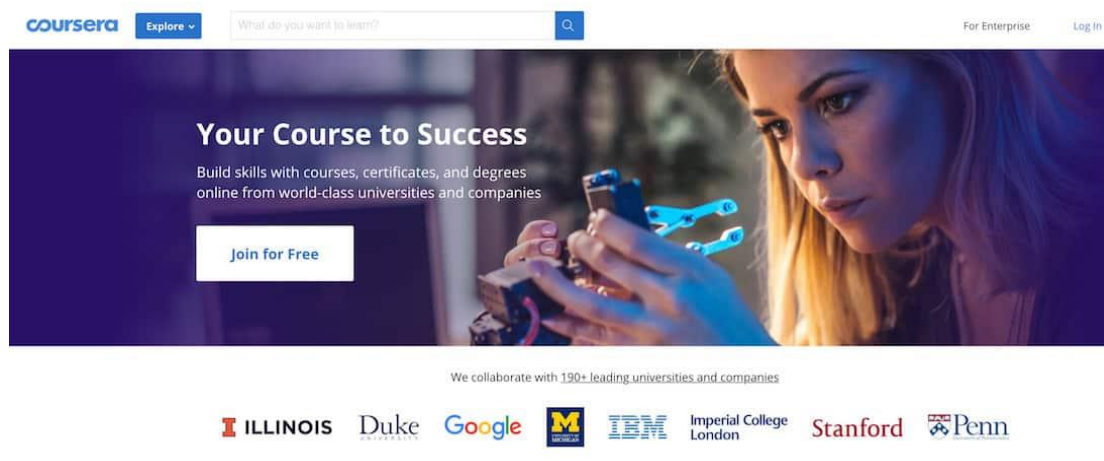


Рисунок 1.3 – Головна сторінка Coursera [2]

Переваги:

1) висока якість навчання. Coursera співпрацює з провідними університетами та навчальними закладами, що гарантує високу якість навчальних матеріалів та інструкцій;

2) сертифікація. Після успішного завершення курсу на Coursera, користувачі отримують сертифікат, який може підтвердити їхні навички та знання відповідної галузі;

3) гнучкість. Користувачі можуть вчитися власним темпом та обирати курси з різних галузей програмування залежно від своїх інтересів та потреб.

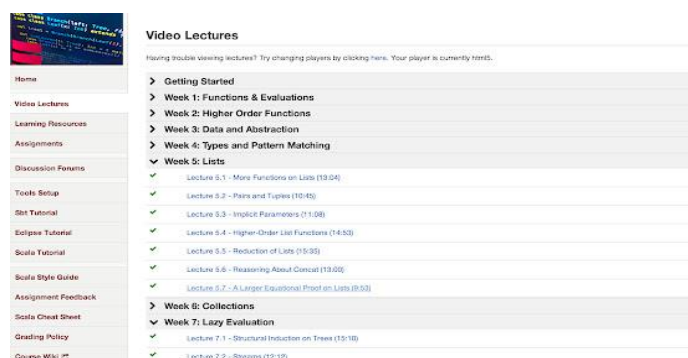


Рисунок 1.4 – Сторінка прогресу курсів [2]

Недоліки:

1) вартість курсів. Деякі курси на Coursera можуть бути досить дорогими, що може утруднити доступ до навчання для деяких користувачів;

2) обмежені можливості. Не всі курси на Coursera можуть бути доступні у будь-який час, і деякі можуть мати обмежену кількість розкладу або недоступність;

3) відсутність інтерактивного навчання. У деяких курсах на Coursera може бути менше можливостей для інтерактивного навчання, порівняно з іншими платформами.

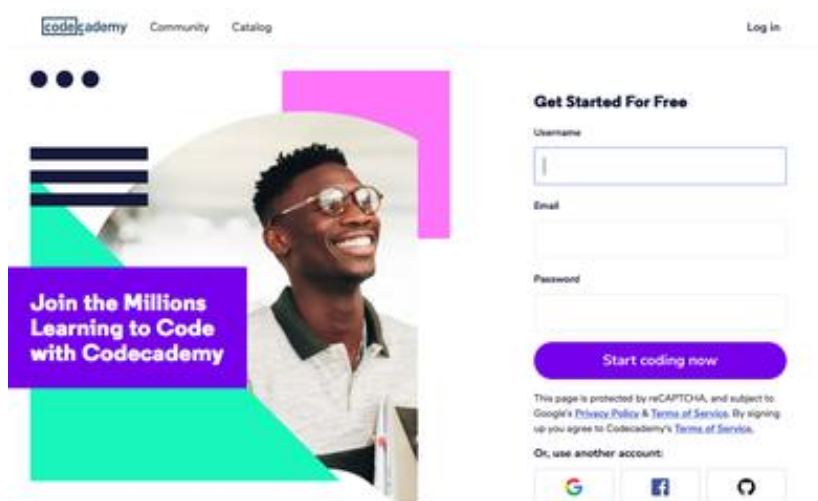


Рисунок 1.5 – Головна сторінка Codecademy [3]

Також варто зазначити платформу Codecademy (рис. 1.5), яка спеціалізується на навчанні програмування через інтерактивні вправи та об'єкти. Codecademy пропонує велику кількість безкоштовних курсів та має активну спільноту користувачів. Однак, вона може бути обмеженою в певних аспектах, таких як вибір тематики курсів (рис. 1.6) та рівень глибини матеріалу.

Переваги:

– інтерактивне навчання. Codecademy пропонує інтерактивні вправи та об'єкти, що допомагають користувачам навчитися програмуванню шляхом практичної роботи;

- безкоштовні курси. Codecademy надає доступ до великої кількості безкоштовних курсів, що робить навчання доступним для багатьох користувачів;
- активна спільнота. Codecademy має активну спільноту користувачів, де можна обмінюватися досвідом, отримувати підтримку та співпрацювати з іншими програмістами.

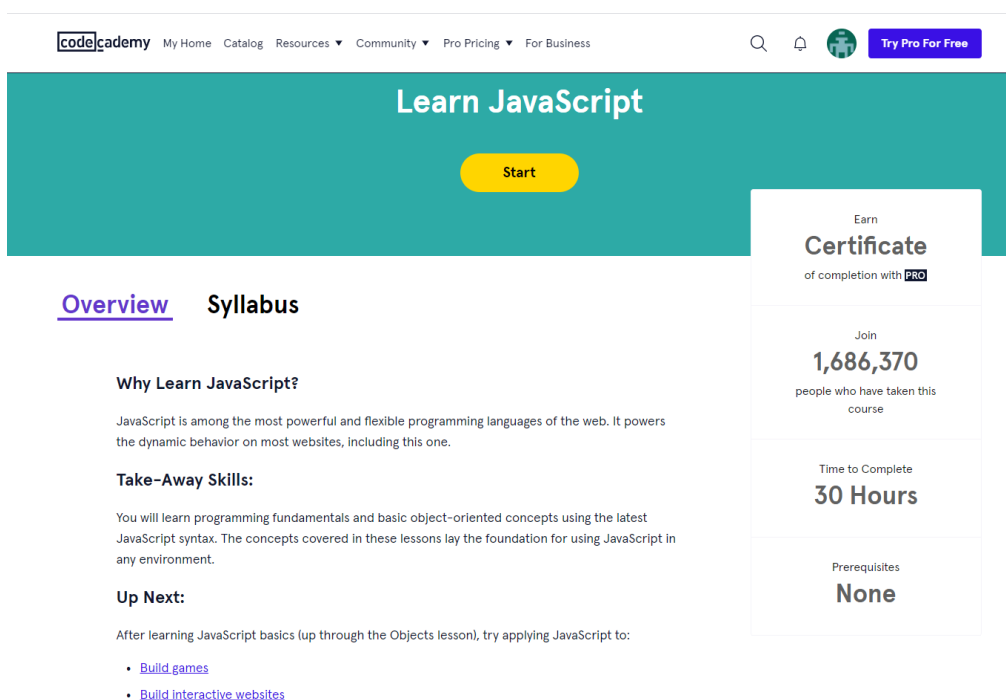


Рисунок 1.6 – Сторінка курсу JavaScript [3]

Недоліки:

- обмеженість тематики. В порівнянні з іншими платформами, Codecademy може мати обмеженішу кількість курсів та тематики, що може не задовольнити всіх потреб користувачів;
- невелика глибина матеріалу. Деякі курси на Codecademy можуть бути менш глибокими або менш детальними в порівнянні з іншими платформами навчання програмування;
- відсутність академічного визнання. Курси на Codecademy не завжди мають академічну вагу або визнання, що може бути важливим для деяких студентів чи професіоналів.

Враховуючи існуючі рішення, метою кваліфікаційної роботи є створення веб-магазину з продажу курсів програмування, який буде комбінувати переваги існуючих платформ та розв'язувати їх недоліки. Головними пріоритетами об'єкту є доступність, різноманітність курсів, якість навчання та зручний інтерфейс для користувачів.

Висновок до розділу 1

У першому розділі кваліфікаційної роботи бакалавра було розглянуто актуальність теми «Розробка веб-магазину з продажу курсів програмування» і визначено мету об'єкту. Також був проведений огляд існуючих рішень, зокрема платформ Udemu, Coursera та Codecademy.

Актуальність теми обумовлена зростанням популярності навчання програмуванню та зростанням попиту на курси з цієї галузі. Розробка веб-магазину з продажу курсів програмування має на меті забезпечити доступність навчання та зручність для студентів, які бажають розвивати свої навички в програмуванні.

Огляд існуючих рішень показав, що на ринку вже присутні платформи, які пропонують курси програмування. Кожна з них має свої переваги та недоліки. Udemu має широкий вибір курсів, але може стикається зі змінною якістю матеріалів. Coursera гарантує високу якість навчання та сертифікацію, але може бути обмеженим в гнучкості та вартості. Codecademy пропонує інтерактивне навчання та безкоштовні курси, але може мати обмежену тематику.

На основі проведеного огляду існуючих рішень та з урахуванням переваг і недоліків, метою кваліфікаційної роботи є розробка веб-магазину з продажу курсів програмування, який комбінує переваги існуючих платформ та розв'язує їх недоліки. Головними пріоритетами об'єкту є доступність, різноманітність курсів, якість навчання та зручний інтерфейс для користувачів.

Дане дослідження є актуальним і важливим, оскільки сприятиме розвитку освіти в галузі програмування та забезпечить користувачам можливість

вивчення та придбання курсів з різних галузей програмування. Далі у роботі буде розглянуто деталі розробки веб-магазину з продажу курсів програмування, включаючи архітектуру системи, вибір технологій та реалізацію функціоналу.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

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

Для початку необхідно дослідити бази даних, та структуру об'єкту, почнемо з системи керування базами даних.

Система керування базами даних, скор. СКБД (англ. Database Management System, скор. DBMS) – сукупність програмних та лінгвістичних засобів загального або спеціального призначення, що забезпечують управління створенням та використанням баз даних.

СКБД – комплекс програм, що дозволяють створити базу даних (БД) та маніпулювати даними (вставляти, оновлювати, видаляти та вибирати). Система забезпечує безпеку, надійність зберігання та цілісність даних, а також надає засоби для адміністрування БД.

Типи СКБД:

- реляційні;
- ключ-значення;
- документні;
- графові;
- колонкові.

Реляційні СКБД – класичні, реляційні СКБД найчастіше використовують для побудови рішень OLTP (Online Transaction Processing). У таких рішеннях СКБД працює з невеликими за розмірами транзакціями, але такими, що йдуть великим потоком, і при цьому від системи потрібен мінімальний час відгуку, а також можливість, за певних умов, скасувати будь-які зміни виконуваних у рамках транзакції. Якщо ви будете систему, в рамках якої потрібно зберігати значну кількість сутностей (таблиць), з різними типами зв'язків між ними (один–до–одного, один–до–багатьох, багато–багатьох), то це швидше за все про реляційні СКБД.

Найбільш відомі СКБД такого типу – Oracle, Microsoft SQL, PostgreSQL, MySQL.

Один із найпростіших типів СКБД. У спрощеному вигляді, це певна таблиця з унікальним ключем і власне пов'язаним з ним значенням, в якому може бути будь-що. Найчастіше такі СКБД застосовують для кешування, т.к. вони дуже швидко працюють, а це і не складно, коли є унікальний ключ і запит повертає тільки одне значення. У деяких представників даних СКБД є можливість працювати повністю в пам'яті, а також є можливість задавати термін життя запису, після закінчення якого записи будуть автоматично видалятися.

Найбільш відомі СКБД такого типу – Redis та Memcached.

Документні або документно – орієнтовані СКБД – це один з найпопулярніших різновидів NoSQL СКБД, де основною одиницею логічної моделі даних є документ – структурований текст, з певним синтаксисом.

Іноді зустрічаються думки, що модель даних у документних БД схожа на модель даних в об'єктно-орієнтованих базах даних. У цьому є частка правди, єдина реальна різниця між ними полягає в тому, що бази даних документів зберігають лише стан, але не поведінку.

Так само сама назва «документно-орієнтована» часом вводить в оману, і мені зустрічалися колеги, які вважали, що це база для систем документообігу. Ні це не так.

Цікаво, що документні СКБД розвиваються досить активно, і зараз деякі з них, зокрема, підтримують перевірку схеми.

Відомими представниками таких СКБД є CouchDB, MongoDB, Amazon DocumentDB.

Графові СКБД – специфічний тип, призначені для роботи з графами, з їх вузлами, властивостями та довільними відносинами між вузлами.

Дуже простий приклад, це організація зв'язків у різного типу соціальних мереж, де потрібно зберігати зв'язки між користувачами (вузлами) за різними критеріями (родинні зв'язки, колеги, спільні інтереси).

Відомі представники цього типу скбд – Neo4j, Amazon Neptune, InfiniteGraph, InfoGrid.

Колонкові СКБД дуже схожі на реляційні. Вони також складаються з рядків, які мають атрибути, а рядки групуються в таблицях. Відмінності в логічних моделях несуттєві, а на рівні фізичного зберігання даних відмінності значні.

У реляційних СКБД дані зберігаються «построчно», це означає, що для зчитування значення певної колонки, доведеться прочитати практично весь рядок, як мінімум від першої до потрібної колонки. У колонковій СКБД дані зберігаються «поколоночно», тобто. колонка – це як окрема таблиця. Відповідно, читання відбуватиметься з конкретного стовпця відразу. Насправді це реально працює дуже швидко (перевірено мною на кількох реалізованих сховищах даних).

Основні переваги колонкових СКБД – ефективне виконання складних аналітичних запитів на великих обсягах, і легка, практично миттєва, зміна структури таблиць з даними, плюс суттєва компресія та стиснення, що дозволяє значно економити місце.

Яскраві представники колонкових СКБД – Sybase IQ (нині SAP IQ), Vertica, ClickHouse, Google BigTable, InfoBright, Cassandra.

При виборі типу СКБД слід, перш за все, виходити з типу розв’язуваних завдань, типів даних, перспектив зростання і масштабування як показано у таблиці 2.1.

Таблиця 2.1 – Типи СКБД

Тип СКБД	Коли обирати	Приклади популярних СКБД
Реляційні	Потрібна транзакційність; висока нормалізація; велика частка операцій на вставку	Oracle, MySQL, Microsoft SQL Server, PostgreSQL
Ключ-значення	Завдання кешування та брокери повідомлень	Redis, Memcached

Продовження таблиці 2.1

Документні	Для зберігання об'єктів у одній сутності, але з різною структурою; зберігання структур на основі JSON	CouchDB, MongoDB, Amazon DocumentDB
Графові	Завдання подібні до соціальних мереж; системи оцінок та рекомендацій	Neo4j, Amazon Neptune, InfiniteGraph, InfoGrid
Колонкові	Сховища даних; вибірки із складними аналітичними обчисленнями; кількість рядків у таблиці перевищує сотні мільйонів	Vertica, ClickHouse, Google BigTable, Sybase \ SAP IQ, InfoBright, Cassandra

2.2 Вибір засобів, методів та алгоритмів вирішення поставленого завдання

Visual Studio Code – це повнофункціональний текстовий редактор для редагування локальних файлів чи бази коду. Він включає різні функції для редагування бази коду, яка допомагає розробникам відстежувати зміни.

Різні функції, що підтримуються в VScode:

- підсвічування синтаксису;
- авто відступ;
- розпізнавання типів файлів;
- бічна панель із файлами вказаного каталогу;
- макрос;
- плагін та пакети.

Vs code використовується як інтегрований редактор розробки (IDE), як Sublime і NetBeans. Поточна версія редактора Vs code сумісна з різними операційними системами, такими як Windows, Linux та MacOS.

Git – це розподілена система контролю версій, яка дозволяє відслідковувати зміни у вихідному коді, керувати ними та співпрацювати у

команді над одним об'єктом. Git використовується в багатьох об'єктах, починаючи від невеликих об'єктів до великих відкритих об'єктів на GitHub.

PNPM – це менеджер пакетів для Node.js, який дозволяє керувати залежностями об'єкту та встановити необхідні модулі за допомогою однієї команди. PNPM пропонує безліч переваг перед стандартним NPM, таких як швидке встановлення, економія місця на жорсткому диску та інші.

ESLint – це інструмент статичного аналізу JavaScript, який дозволяє перевірити код на відповідність правилам та стандартам написання коду. ESLint використовується для підвищення якості коду, спрощення його читання та підтримки. ESLint може використовуватися як за допомогою командного рядка, так і як плагін у редакторі коду.

Тепер розглянемо кожен інструмент докладніше:

Git – це інструмент, який дозволяє відстежувати зміни у вихідному коді та співпрацювати у команді над одним об'єктом. Git дозволяє контролювати версії коду, створювати гілки, об'єднувати їх, відкочувати зміни та багато іншого. Він використовується у більшості об'єктів з відкритим вихідним кодом, а також у комерційних об'єктах. Git дозволяє прискорити процес розробки, уникнути помилок при зміні коду та знизити ймовірність конфліктів під час спільної роботи над об'єктом.

PNPM – це менеджер пакетів для Node.js, який дозволяє керувати залежностями об'єкту та встановити необхідні модулі за допомогою однієї команди. PNPM встановлює кожен модуль в окрему папку, що дозволяє заощаджувати місце на жорсткому диску та прискорює процес встановлення модулів. Крім того, PNPM дозволяє легко перемикатися між версіями модулів та відкочуватися до попередніх версій. Він також може використовуватися як заміна стандартному NPM.

ESLint – це інструмент статичного аналізу JavaScript, який дозволяє перевіряти код на відповідність певним правилам та стандартам. Він дозволяє уникнути помилок, підвищити читання коду, а також зробити його більш підтримуваним та безпечним.

Однією з основних функцій ESLint є перевірка стилю коду. Він може виявляти та виправляти неправильне форматування коду, наприклад, порушення відступів, зайві прогалини, неправильне використання лапок тощо. Також, він перевіряє наявність деяких небезпечних або небажаних шаблонів коду, таких як змінні, що не використовуються, використання заборонених функцій і т.д.

ESLint є дуже гнучким інструментом і дозволяє налаштовувати правила аналізу відповідно до вимог об'єкту та команди розробників. Це можна зробити за допомогою файлів конфігурації `.eslintrc` або `package.json`.

Крім того, ESLint інтегрується з різними IDE, текстовими редакторами та системами складання об'єктів, що дозволяє виконувати перевірку коду на льоту в реальному часі та прискорювати процес розробки.

2.3 Обрання мов програмування

Для створення об'єкту було обрано такі мови програмування: HTML, TAILWIND, POSTCSS, JS, NODE.

HTML – це стандартна мова розмітки веб-сторінок. Він був використаний для створення структури та розмітки веб-сторінки.

TAILWIND – це CSS – фреймворк, який дозволяє швидко і легко створювати інтерфейс користувача. Він був використаний для стилізації веб-сторінки.

POSTCSS – інструмент обробки CSS, який дозволяє використовувати останні версії CSS, а також додавати плагіни для розширення функціональності. Він був використаний для оптимізації та обробки CSS – стилів.

JS – це мова програмування, яка використовується для додавання динамічної поведінки на веб-сторінки. Він був використаний для написання скриптів на стороні клієнта.

Node.js – це середовище виконання JavaScript, яке дозволяє запускати JavaScript – код на стороні сервера. Воно побудоване на двигуні V8, який є основою браузера Google Chrome. Node.js дозволяє розробникам писати

серверні застосунки на JavaScript, що дозволяє реалізувати повний стек технологій на одній мові програмування. Node.js є популярним серед інтернет-розробників і застосовується для створення веб-серверів, API, чат-ботів, мікросервісів, інтерактивних додатків, і багатьох інших застосувань.

Використання цих технологій дозволяє покращити візуальну привабливість, функціональність та інтерактивність інформаційної системи програми лояльності магазину, створюючи захоплююче та зручне користувацьке середовище.

2.4 Структура об'єкта

Розробка IT – об'єкта включає наступні етапи:

- планування. Грамотне планування функціональності майбутнього продукту та аналіз вимог відіграють ключову роль для всього об'єкта;
- дизайн(ui\ux). UX відповідає за функціональні можливості, а UI – за візуальну частину;
- розробка. Робота спринтами (agile) з ітеративним підходом, поділяючи об'єкт на дрібні завдання;
- тестування. Перевірка продукту на готовність до релізу, виконання модульного та інтеграційного тестування;
- підтримка. Готовий продукт може потребувати додаткової підтримки, будь то додаткові питання щодо роботи продукту від клієнтів, або необхідність внести зміни до вже закладених функцій.

Ризики об'єкту:

- В даний час управління ризиками є одним з найбільш актуальних та динамічних напрямів стратегічного та оперативного менеджменту;
- Вважається, що якісне управління ризиками дозволяє використовувати оптимальні за ефективністю та витратами кошти контролю ризиків, що відповідають поточним цілям та завданням. У таблиці 2.2 розглядаються ризики об'єкту.

Таблиця 2.2 – Ризики проекту

п\п	Ризик	Робота\ресурс	Стратегія запобігання ризику	План реакції
Ресурсні ризики				
1	Використання недосвідченого розробника	Розробник	Залучення на допомогу досвідчених працівників; Навчання недосвідчених співробітників	Призначити у допомогу досвідченого співробітника; Призначити навчання
2	Ресурси з великим обсягом робіт	Розробник	Розробка та складання плану, за можливостями співробітника; погодження термінів виконання	Залучення додаткових співробітників для швидшої реалізації
Календарні ризики				
1	Завдання з попередніми тривалостями	Написання коду програми	Залучення досвідченого співробітника для реалізації завдання у строк	Додати до допомоги більш досвідченого співробітника
2	Надто короткі завдання	Усунення виявлених невідповідностей	Додавання завдань, які запобігають невідповідності	Додати до плану завдання: «Функціональне тестування»

Джерело: складено автором

План реакції на ризик:

– Для усунення ризиків усунення виявлених невідповідностей, було додано додаткове завдання «Функціональне тестування». Цей розділ допоможе запобігти помилкам, що виникли в результаті розробки об'єкту, завдяки чому потрібна мінімальна кількість часу на етап «Усунення виявлених невідповідностей»;

– Ризики, пов'язані з ресурсами, запобігти заздалегідь можна шляхом призначення навчання ще до початку об'єкта. Завдяки цьому ми підвищимо рівень кваліфікованості персоналу, а також забезпечимо закриття об'єкта вчасно.

Висновки до розділу 2

У другому розділі кваліфікаційної роботи бакалавра проведено опис основних методів та технологій об'єкта, а також обґрунтування вибору засобів реалізації. Зокрема, було розглянуто питання обрання системи керування базами даних СКБД, середовища розробки і мови програмування.

При обранні СКБД було враховано вимоги об'єкту та його потенційні потреби щодо зберігання та обробки даних. Після аналізу різних варіантів СКБД було прийнято рішення використати NoSql, оскільки вона відповідає потребам об'єкту і має високу надійність та широкий функціонал.

У контексті обрання середовища розробки було проаналізовано різні варіанти і визначено Visual Studio як найбільш підходяще для реалізації веб-магазину. Це середовище надає потрібні інструменти та можливості для зручної розробки, налагодження та тестування програмного коду.

Також було проведено аналіз різних мов програмування і визначено, що Java Script є найбільш підходящою для реалізації веб-магазину з продажу курсів програмування. Ця мова має широку підтримку, багатофункціональність та дозволяє ефективно вирішувати поставлені завдання об'єкту.

Отже, результати розділу підтверджують відповідність обраних методів та технологій потребам та цілям розробки веб-магазину з продажу курсів програмування. Ці обрані засоби реалізації створять надійну та функціональну платформу для користувачів та адміністраторів. У наступному розділі будуть детальніше розглянуті планування та реалізація об'єкта з використанням обраних методів та технологій.

РОЗДІЛ 3

РОЗРОБКА ВЕБ ДОДАТКУ

3.1 Практична реалізація об'єкта проектування

У третьому розділі кваліфікаційної роботи бакалавра проведено детальний аналіз вимог до програмного продукту – веб-магазину з продажу курсів програмування. Аналіз вимог є важливим етапом у розробці будь-якого програмного продукту, оскільки дозволяє зрозуміти потреби та очікування користувачів, а також встановити функціональні та нефункціональні вимоги, якими повинен відповідати продукт.

3.1.1 Визначення функціональних вимог

Функціональні вимоги визначають конкретні функції та можливості, які повинен мати веб-магазин. Під час аналізу були виявлені наступні функціональні вимоги:

- авторизація користувачів. Користувачі повинні мати можливість авторизуватися в системі, використовуючи свої облікові дані;
- перегляд каталогу курсів. Користувачам потрібно мати можливість переглядати доступні курси з детальним описом, ціною, автором тощо;
- додавання курсів до кошика. Користувачі повинні мати можливість додавати обрані курси до свого кошика для подальшої покупки;
- оформлення замовлення. Після додавання курсів до кошика користувачі повинні мати можливість оформити замовлення, вказати необхідну інформацію для доставки та сплати;
- оплата курсів. Система повинна підтримувати різні методи оплати, наприклад, кредитні картки, електронні платіжні системи тощо;
- можливість перегляду придбаних курсів. Після успішної оплати користувачі повинні мати доступ до придбаних курсів та можливість переглядати їх в будь-який зручний час.

3.1.2 Визначення нефункціональних вимог

Нефункціональні вимоги визначають характеристики, які повинен мати програмний продукт, такі як продуктивність, безпека, надійність та інші. Під час аналізу були виявлені наступні нефункціональні вимоги:

- продуктивність. Веб-магазин повинен працювати швидко та ефективно, забезпечуючи короткий час завантаження сторінок та оперативну обробку запитів користувачів;
- безпека. Система повинна забезпечувати захист персональних даних користувачів, використовуючи шифрування та інші заходи безпеки;
- надійність. Веб-магазин повинен працювати стабільно та надійно, запобігаючи випадковим збоям та відновлюючи роботу після виникнення помилок;
- сумісність. Система повинна бути сумісною з різними веб-браузерами та пристроями, забезпечуючи однакову функціональність та зручне відображення на різних платформах;
- інтерфейс. Користувацький інтерфейс веб-магазину повинен бути зрозумілим, зручним у використанні та естетично здоровим;
- масштабованість. Система повинна бути готовою до масштабування, здатною обробляти зростаючий обсяг даних та користувацьких запитів без втрати продуктивності.

Аналіз вимог до програмного продукту є важливим кроком, що допомагає зрозуміти потреби користувачів та встановити основу для подальшої розробки. Отримані функціональні та нефункціональні вимоги будуть використовуватись у наступних розділах для планування та реалізації програмного продукту.

3.2 Визначення компонентів програми

Під час проектування було визначено основні компоненти програми, які включають:

- клієнтський інтерфейс. Цей компонент відповідає за візуальну частину веб-магазину. Він включає в себе розміщення та оформлення елементів сторінок, взаємодію з користувачем та передачу запитів на сервер;
- серверний компонент. Цей компонент відповідає за обробку запитів користувачів, виконання бізнес-логіки та взаємодію з базою даних. Він забезпечує передачу необхідної інформації між клієнтським інтерфейсом та базою даних;
- база даних. Цей компонент забезпечує зберігання і організацію даних про курси, користувачів, замовлення та іншу інформацію, необхідну для роботи веб-магазину. Він включає в себе таблиці, структури даних та механізми доступу до них як представлено на рисунку 3.1.

3.2.1 Визначення модулів та їх взаємодії

Після визначення компонентів програми було розроблено структуру модулів та взаємодію між ними. Основні модулі, що були ідентифіковані, включають:

- модуль авторизації та аутентифікації. Цей модуль відповідає за реєстрацію, авторизацію та перевірку ідентичності користувачів. Він забезпечує безпеку та конфіденційність даних користувачів;
- модуль каталогу курсів. Цей модуль відповідає за відображення каталогу курсів, їх пошук, сортування та фільтрацію. Він забезпечує зручний спосіб перегляду та вибору курсів для користувачів;
- модуль кошика. Цей модуль дозволяє користувачам додавати курси до кошика, переглядати його вміст, редагувати кількість курсів та оформляти замовлення;
- модуль оплати. Цей модуль відповідає за обробку платежів та забезпечує безпечну оплату курсів через різні платіжні системи або кредитні картки;
- модуль управління замовленнями. Цей модуль дозволяє адміністраторам переглядати, опрацьовувати та відстежувати стан замовлень, а також забезпечує взаємодію зі службами доставки.

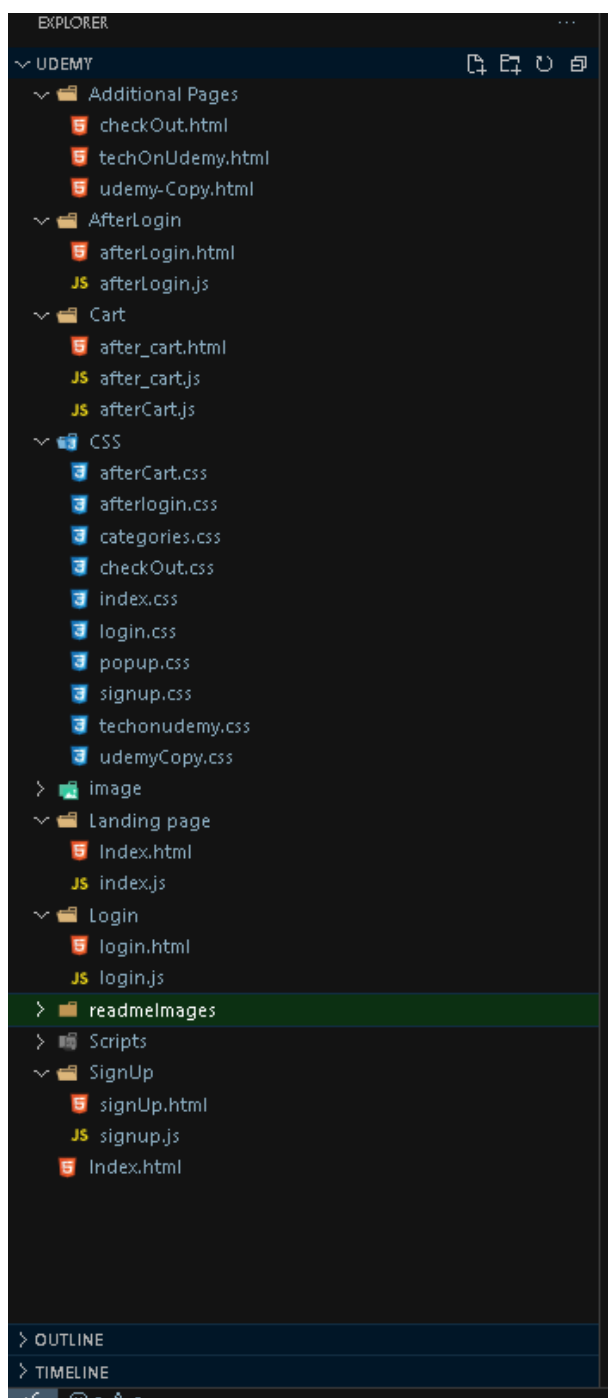


Рисунок 3.1 – Файлова структура об’єкта

Файлова структура об’єкту наведена на рисунку 3.1. Крім того, були визначені механізми взаємодії між модулями, такі як передача даних, виклики функцій та обробка подій.

3.2.2 Визначення архітектурного шаблону

Для реалізації структури програми був обраний архітектурний шаблон MVC (Model– View– Controller). Цей шаблон, зображено на рисунку 3.2, розділяє програму на три компоненти: модель, яка представляє дані та бізнес–

логіку; вид, який відповідає за візуальне відображення інформації; та контролер, який керує взаємодією між моделлю та видом.

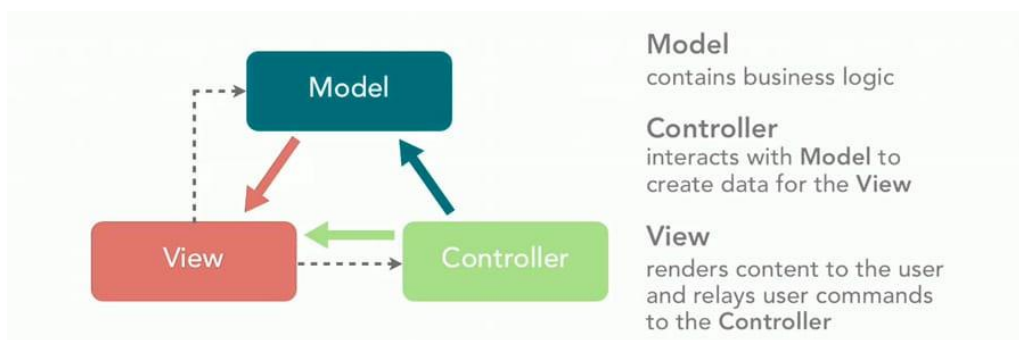


Рисунок 3.2 – Паттерн MVC

Архітектурний шаблон MVC дозволить забезпечити модульність, розширюваність та підтримку принципу однорідності коду. Він також сприятиме подальшому розвитку та підтримці програмного продукту.

Також було проведено проектування структури програми на основі визначених компонентів, модулів та взаємодії між ними. Архітектурний шаблон MVC був обраний для забезпечення ефективної організації програмного продукту. Отримана структура програми буде використана в наступних розділах для розробки та реалізації функціональності веб-магазину.

3.3 Розробка та тестування

Цей етап включає реалізацію функціональності програмного продукту на основі визначених вимог та проектування структури програми.

3.3.1 Реалізація функціональності

Під час розробки було проведено програмування компонентів та модулів, відповідно до вимог до програмного продукту. За допомогою вибраних мов програмування та середовища розробки було написано необхідний код для реалізації функціональності веб-магазину.

Наприклад, для реалізації модуля авторизації та аутентифікації було розроблено механізм реєстрації користувачів, перевірку ідентичності та

збереження даних авторизації. Для модуля каталогу курсів було реалізовано функції відображення курсів, пошуку, сортування та фільтрації. Аналогічно, для модуля кошика були розроблені функції додавання, видалення та зміни кількості курсів у кошику.

3.3.2 Тестування

Після реалізації функціональності було проведено тестування програмного продукту. Тестування є важливим кроком для перевірки правильності реалізації та виявлення можливих помилок або недоліків. У цьому процесі були використані різні методи та підходи, такі як модульне тестування, інтеграційне тестування та системне тестування.

Модульне тестування дозволяє перевірити окремі функціональні блоки або модулі програмного продукту на коректність їх роботи. Інтеграційне тестування використовується для перевірки взаємодії між компонентами та модулями, а системне тестування спрямоване на перевірку роботи всього програмного продукту як єдиного цілого.

Під час тестування були виявлені та виправлені деякі помилки, а також перевірена відповідність функціональності вимогам до програмного продукту. Тестові сценарії були розроблені для охоплення різних ситуацій та перевірки роботи різних функціональних можливостей.

Усі результати тестування та виправлення помилок були документовані для подальшої використання та покращення програмного продукту. Було успішно реалізовано функціональність веб-магазину та проведено відповідне тестування. Результати цього етапу будуть використані у наступних розділах для розгляду результатів та вдосконалення програмного продукту.

3.4 Встановлення необхідної конфігурації

У розділі проведено встановлення необхідної конфігурації для належної роботи веб-магазину з продажу курсів програмування. Цей етап включає

налаштування серверного середовища, бази даних та інших компонентів, необхідних для роботи програмного продукту.

1) встановлення середовища розробки:

- встановіть текстовий редактор, такий як Visual Studio Code (рис. 3.3), Sublime Text або інший за вашим вибором;

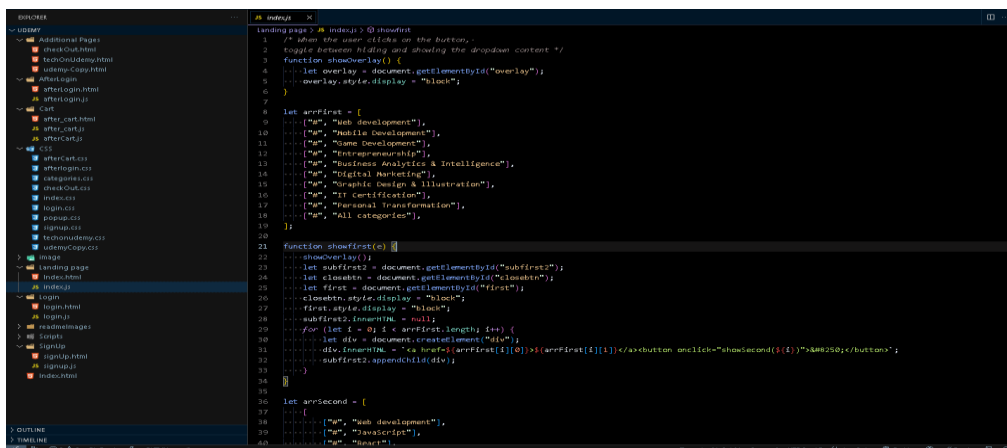


Рисунок 3.3 – Інтерфейс Visual Studio Code

Впевніться, що на вашому комп'ютері встановлено останню версію браузера, наприклад Google Chrome або Mozilla Firefox, для перевірки та тестування веб-сторінок;

2) встановлення потрібних технологій та бібліотек:

- встановіть актуальну версію Node.js (<https://nodejs.org>), зображено на рисунку 3.4 для використання пакетного менеджера npm;

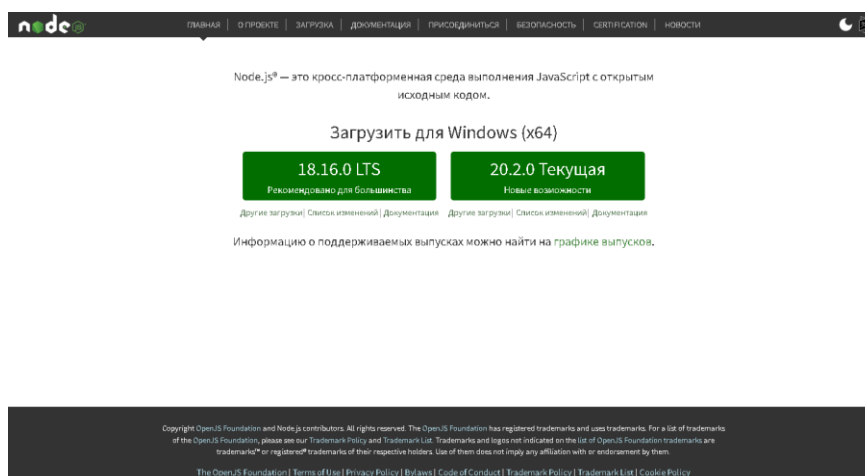


Рисунок 3.4 – Головна сторінка Node.js [4]

- за допомогою командного рядка або терміналу перейдіть до папки об'єкту і ініціалізуйте об'єкт командою `npm init` (рис. 3.5), що створить файл `package.json`, де будуть зберігатись залежності об'єкту.
- встановіть необхідні пакети та бібліотеки для розробки сайту-візитки.

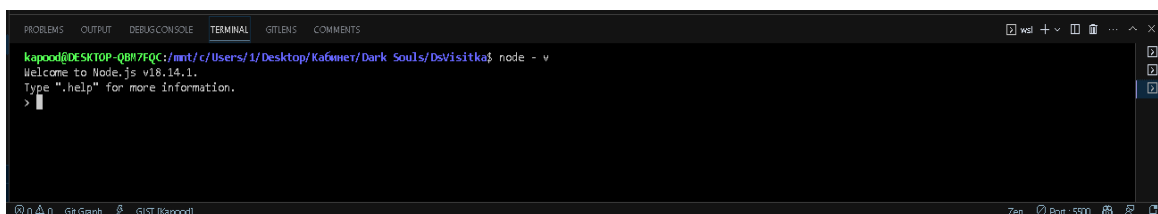


Рисунок 3.5 – Перевірка версії Node.js

Наприклад:

- для роботи з HTML, CSS і JavaScript: немає необхідності встановлювати додаткові пакети, оскільки ці технології входять в стандартний набір веб-браузерів;

- POSTCSS: `npm install postcss;`
- TAILWIND CSS: `npm install tailwindcss.`

3) перевірка конфігурації:

- запустіть локальний сервер для перевірки та тестування розробленого сайту, якщо все встановлено правильно об'єкт буде запущено на порту 5500, як зазначено на рисунку 3.6;



Рисунок 3.6 – Перевірка запуску

- відкрийте веб-браузер та перевірте, чи правильно відображаються сторінки, чи коректно працюють інтерактивні елементи, чи відображаються 3D-графіка, зображення та відео, чи виконується стилізація за допомогою TAILWIND CSS тощо;

— впевнитися, що всі функціональні вимоги до програмного продукту виконуються належним чином, і якщо все встановлено вірно, ви побачите результат виконання, який наведено на рисунку 3.7.

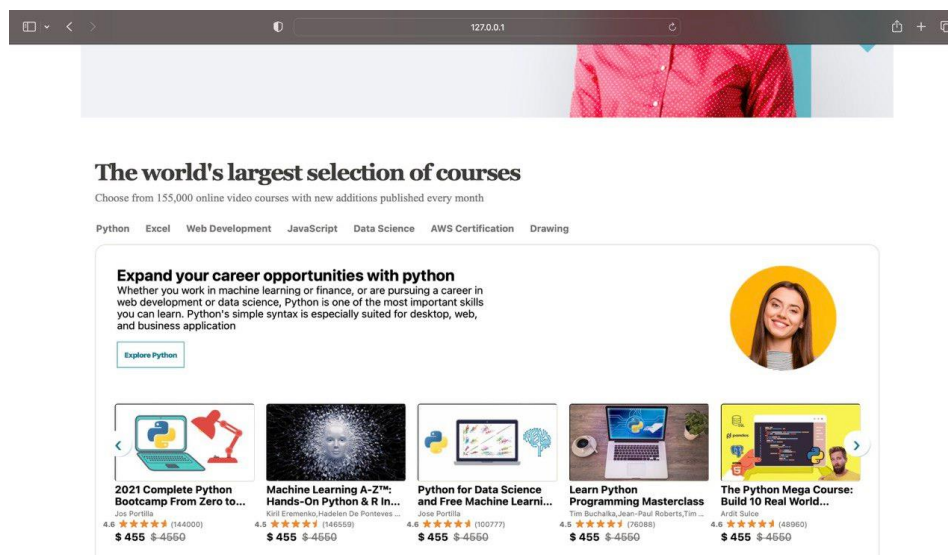


Рисунок 3.7 – Коректне відображення сторінки

Після успішного встановлення необхідної конфігурації ви готові до продовження розробки та тестування веб-додатку.

3.5 Опис і реалізація основних компонентів

3.5.1 Модуль Sign Up

Модуль Sign Up веб-магазину з продажу курсів програмування є важливою складовою частиною системи, оскільки він забезпечує можливість реєстрації нових користувачів. Рисунок 3.8, демонструє візуалізацію модулю, також нижче представлено опис плюсів та мінусів використання цього модулю в кваліфікаційній роботі.

Переваги модуля Sign Up:

— зручна реєстрація. Модуль Sign Up надає можливість користувачам зареєструватися в системі швидко і зручно. Він включає поля для введення повного імені, електронної пошти та пароля;

Рисунок 3.8 – Реалізація Sing up

— перевірка надійності пароля. Модуль має функціонал, який оцінює надійність введеного пароля і надає зворотний зв'язок користувачу щодо сили його пароля. Це допомагає покращити безпеку веб-магазину, стимулюючи користувачів використовувати складніші паролі;

— зберігання даних користувачів. Модуль Sign Up дозволяє зберігати дані нових користувачів, такі як повне ім'я, електронна пошта та пароль. Ці дані можуть використовуватися для автентифікації та виконання інших операцій у системі.

Недоліки модуля Sign Up:

— відсутність підтвердження електронної пошти. Модуль Sign Up не включає механізм підтвердження електронної пошти, що може призвести до можливості реєстрації користувачів з недійсними або фальшивими електронними адресами. Це може вплинути на достовірність даних користувачів у системі;

— обмежена перевірка повного імені. Модуль Sign Up здійснює лише просту перевірку наявності пробілу в повному імені користувача. Це може

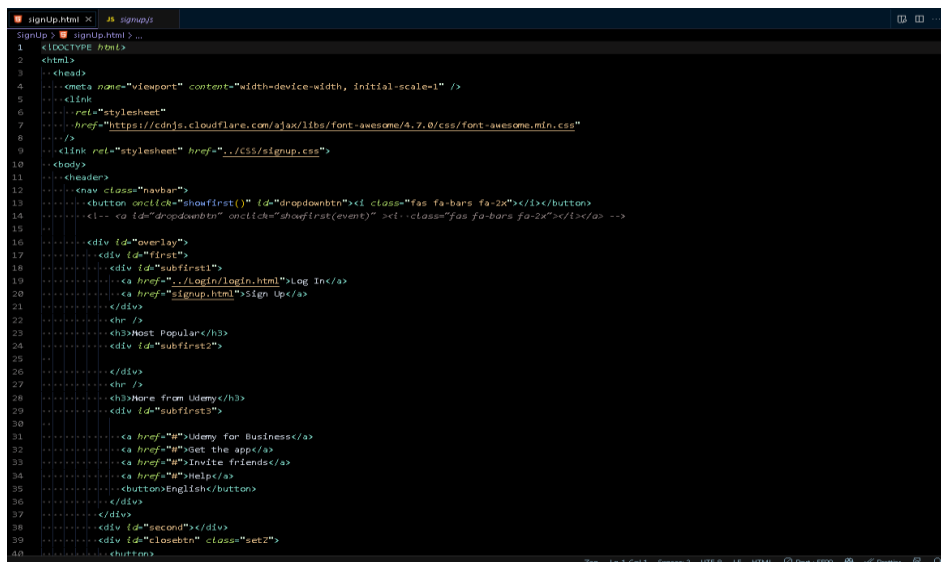


Рисунок 3.10 – Html структура

Загалом, модуль Sign Up є важливою частиною кваліфікаційної роботи і надає користувачам зручну можливість реєстрації веб-магазину. Втім, для покращення безпеки та достовірності даних, рекомендується вдосконалити механізми перевірки електронної пошти та повного імені користувача.

3.5.2 Модуль LandingPage

Модуль LandingPage (рис. 3.11) є важливою складовою кваліфікаційної роботи і відіграє ключову роль у презентації та візуалізації доступних категорій курсів для користувачів. У цьому розділі будуть розглянуті переваги використання модуля LandingPage, його основні переваги та аспекти.

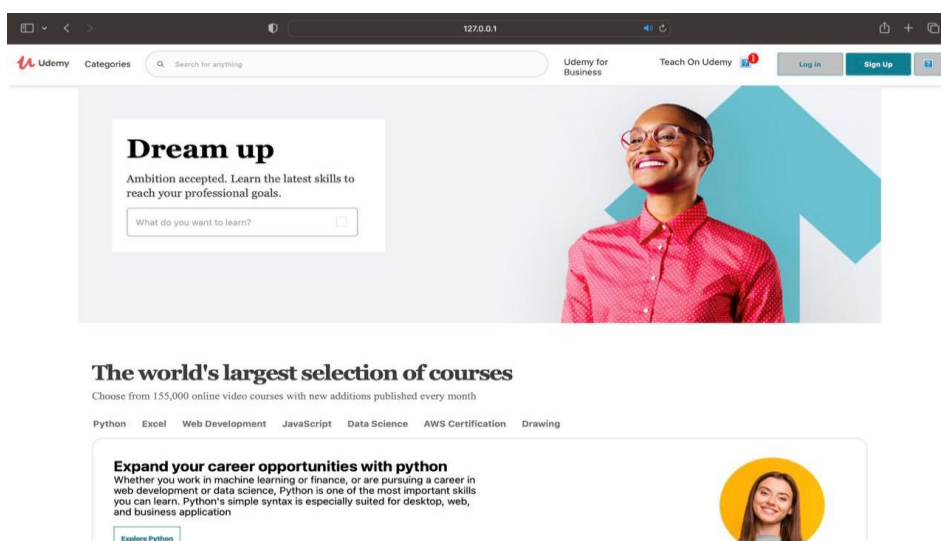


Рисунок 3.11 – Основна сторінка

Переваги використання модуля LandingPage:

- інтуїтивний і зрозумілий інтерфейс. Модуль LandingPage надає користувачам зручний та привабливий інтерфейс, який дозволяє швидко зорієнтуватися у доступних категоріях курсів програмування. Інтуїтивність і зрозумілість інтерфейсу дозволяють користувачам легко орієнтуватися та вибирати курси, які їх зацікавили;
- покращена навігація. Завдяки модулю LandingPage, користувачам легко переходити від однієї категорії до іншої, забезпечуючи зручну навігацію по веб-магазину. Модуль LandingPage може містити графічні елементи, які вказують на доступні категорії курсів, а також забезпечувати посилання на сторінки кожної категорії для отримання додаткової інформації;
- вибір курсів за інтересами. Модуль LandingPage надає можливість користувачам вибрати категорію курсу відповідно до їхніх інтересів і потреб. Це допомагає забезпечити персоналізований підхід до вибору курсу і підвищує шанси на задоволення користувача;
- вручне відображення додаткової інформації. Модуль LandingPage може містити короткий опис кожної категорії курсу (рис. 3.12), що дозволяє користувачам отримати загальне уявлення про доступні напрямки навчання. Це сприяє швидкому ознайомленню користувачів з контентом та може підвищити їх інтерес до певного курсу.

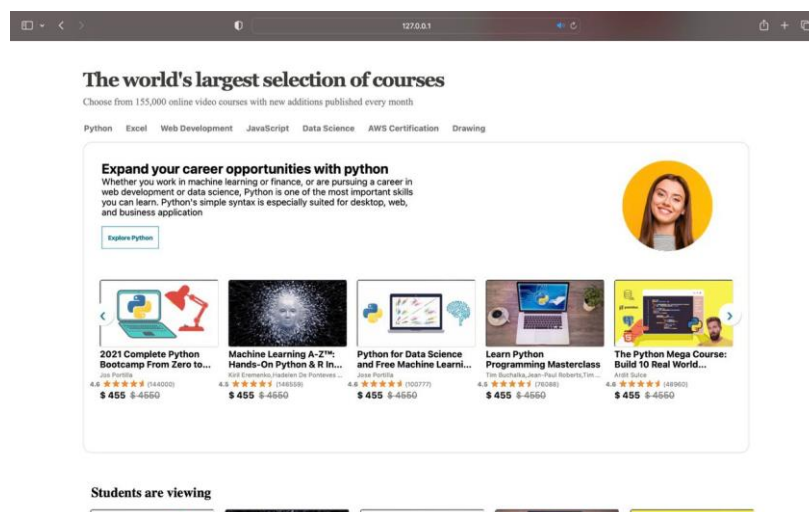


Рисунок 3.12 – Огляд курсів

Основні функції модуля LandingPage:

- дизайн і структура. Важливо розробити ефективний та привабливий дизайн модуля LandingPage (рис. 3.13), який буде гармонійно поєднуватися з дизайном загального веб-магазину. Структура модуля повинна бути логічною та легко зрозумілою для користувачів;
- актуалізація та підтримка. Модуль LandingPage потребує регулярної актуалізації та підтримки для забезпечення актуальності та відображення нових курсів. Важливо планувати час для оновлення і додавання нового контенту на сторінку LandingPage;
- реактивність і адаптивність. Модуль LandingPage повинен бути реактивним і адаптивним до різних типів пристроїв, таких як комп'ютери, планшети та смартфони. Це дозволить користувачам зручно переглядати та вибирати курси незалежно від пристрою, який вони використовують;
- seo – оптимізація. Важливо враховувати принципи SEO – оптимізації при розробці модуля LandingPage. Це включає в себе використання ключових слів, оптимізацію заголовків та метатегів, а також використання правильних URL – адрес для поліпшення видимості веб-магазину в пошукових системах.

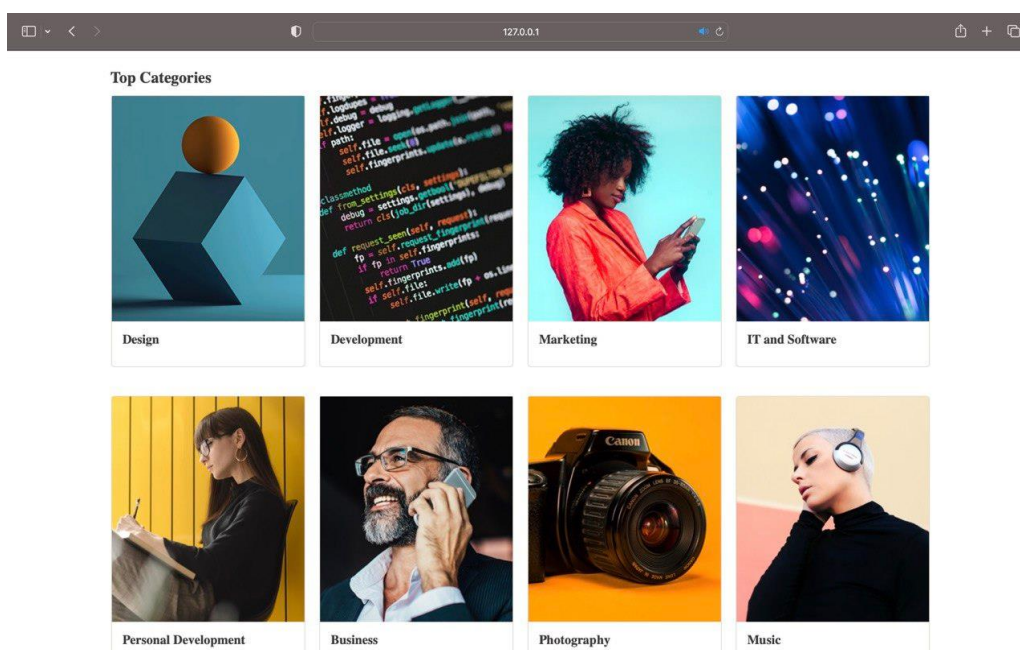
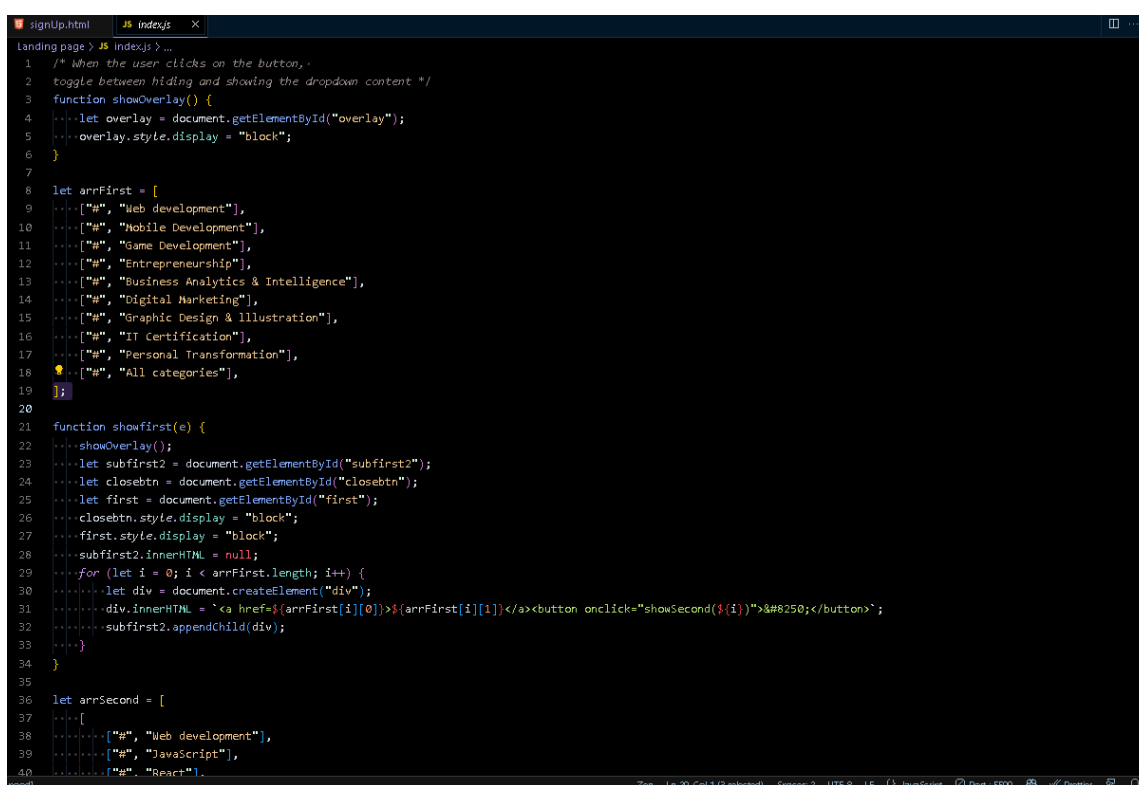


Рисунок 3.13 – Топ категорії

Основні функції для модуля LandingPage, які наведено на рисунку 3.14, включають:

- візуалізація категорій курсів. Основна функція LandingPage полягає в відображенні доступних категорій курсів програмування. Це може бути представлено у вигляді відповідних зображень, іконок або назв категорій. Головна мета полягає в тому, щоб користувачі могли швидко отримати уявлення про різні напрямки навчання, які пропонуються веб-магазином;



```

landing page > JS index.js > ...
1  /* When the user clicks on the button,
2  toggle between hiding and showing the dropdown content */
3  function showOverlay() {
4      let overlay = document.getElementById("overlay");
5      overlay.style.display = "block";
6  }
7
8  let arrFirst = [
9      ["#", "Web development"],
10     ["#", "Mobile Development"],
11     ["#", "Game Development"],
12     ["#", "Entrepreneurship"],
13     ["#", "Business Analytics & Intelligence"],
14     ["#", "Digital Marketing"],
15     ["#", "Graphic Design & Illustration"],
16     ["#", "IT Certification"],
17     ["#", "Personal Transformation"],
18     ["#", "All categories"],
19 ];
20
21 function showfirst(e) {
22     showOverlay();
23     let subfirst2 = document.getElementById("subfirst2");
24     let closebtn = document.getElementById("closebtn");
25     let first = document.getElementById("first");
26     closebtn.style.display = "block";
27     first.style.display = "block";
28     subfirst2.innerHTML = null;
29     for (let i = 0; i < arrFirst.length; i++) {
30         let div = document.createElement("div");
31         div.innerHTML = `<a href=${arrFirst[i][0]}>${arrFirst[i][1]}</a><button onclick="showSecond(${i})">#8250</button>`;
32         subfirst2.appendChild(div);
33     }
34 }
35
36 let arrSecond = [
37     [
38         ["#", "Web development"],
39         ["#", "JavaScript"],
40         ["#", "React"]

```

Рисунок 3.14 – Приклад реалізації основних функцій

- інтуїтивний інтерфейс користувача. LandingPage повинен мати простий, зрозумілий та легкий у використанні інтерфейс. Користувачам слід легко розуміти, як навігуватися по сторінці, вибрати категорію курсу та перейти до детальнішої інформації. Інтерактивні елементи, такі як кнопки або посилання, можуть використовуватися для полегшення процесу переходу;

- персоналізація вибору курсів. Одна з важливих функцій LandingPage — надання можливості користувачам персоналізувати вибір курсів за їхніми інтересами або потребами. Це може бути досягнуто шляхом

відображення рекомендацій або фільтрів, які дозволяють користувачам відбирати курси залежно від певних параметрів, таких як рівень складності, мова програмування тощо;

- додаткова інформація про курси. LandingPage може також надавати короткий опис кожної категорії курсу, щоб користувачі могли отримати загальне уявлення про те, що включається в кожному напрямку навчання. Це може включати опис тематики, навички, які будуть розвиватися, і короткі відомості про викладачів або експертів, які проводять курс;

- покращена навігація. LandingPage повинен забезпечувати зручну навігацію користувачам. Це означає, що користувачі повинні мати можливість легко переходити від однієї категорії курсу до іншої, повертатися на головну сторінку або переходити до детальної інформації про конкретний курс. Меню або навігаційні панелі можуть використовуватися для забезпечення легкого доступу до різних розділів сайту;

- відповідність мобільним пристроям. З урахуванням зростання використання мобільних пристроїв, важливо забезпечити, щоб LandingPage був адаптивним і добре відображався на різних типах пристроїв і розмірах екранів. Це дозволить користувачам зручно переглядати категорії курсів та отримувати доступ до інформації незалежно від пристрою, який вони використовують;

- seo – оптимізація. Важливо враховувати SEO – аспекти при розробці LandingPage. Це означає використання відповідних метатегів, ключових слів, описів та оптимізацію швидкості завантаження сторінки. SEO – оптимізований LandingPage допоможе покращити видимість веб-магазину в пошукових системах і залучити більше потенційних користувачів.

Загалом, модуль LandingPage є важливим інструментом для презентації курсів програмування веб-магазину. Його переваги, такі як інтуїтивний інтерфейс, поліпшена навігація, вибір курсів за інтересами та зручне відображення інформації, допомагають залучити та задовольнити користувачів. При правильному дизайні, актуалізації, реактивності, адаптивності та SEO –

оптимізації модуль LandingPage може стати потужним інструментом продажу курсів програмування веб– магазину.

3.5.3 Модуль Cart

Модуль Cart (рис. 3.15) в кваліфікаційній відіграє важливу роль у зручному зборі та оформленні замовлень користувачів. Цей модуль дозволяє користувачам додавати курси до кошика, керувати кількістю курсів та остаточними вартостями, а також здійснювати оформлення замовлення.

Описавши переваги та недоліки використання модуля Cart, можна виділити наступні основні переваги та аспекти.

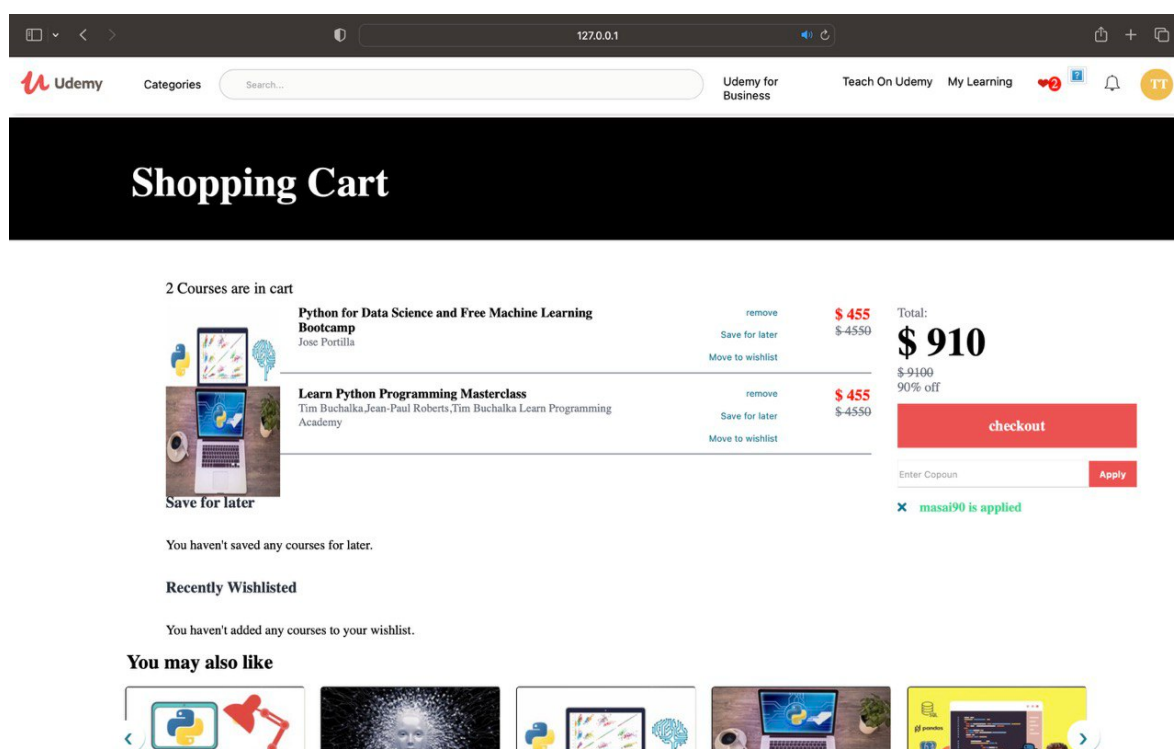
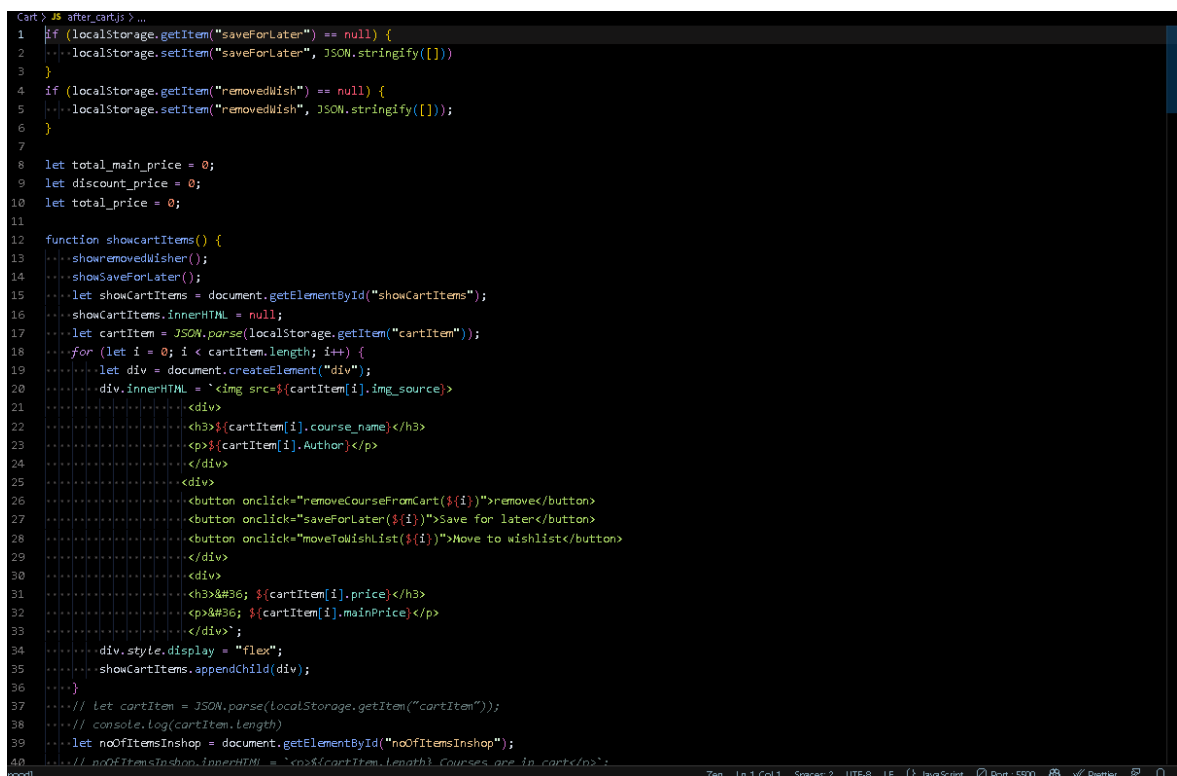


Рисунок 3.15 – Реалізація модулю Cart

Переваги використання модуля Cart:

– зручне зберігання та управління товарами. Модуль Cart (рис. 3.16) дозволяє користувачам додавати обрані курси до кошика, зберігаючи їх для подальшої покупки. Користувачам пропонується можливість змінювати кількість курсів або видаляти їх з кошика за їхніми потребами. Це забезпечує зручний спосіб керування вибраними товарами перед оформленням замовлення;

- обчислення остаточної вартості. Модуль Cart автоматично обчислює остаточну вартість замовлення на основі вибраних курсів та їх кількості. Це дозволяє користувачам з легкістю визначати загальну вартість покупки перед переходом до оформлення замовлення. Такий підхід допомагає користувачам планувати свої витрати та приймати інформовані рішення;
- зручність оформлення замовлення. Модуль Cart надає користувачам зручний і простий процес оформлення замовлення. Користувачам пропонується заповнити необхідні дані, такі як адреса доставки та інформація про платіж, для завершення процесу покупки. Це дозволяє зменшити кількість кроків та спростити процес оформлення замовлення для забезпечення зручності користувачів;



```

Cart > after_cart.js > ...
1  if (localStorage.getItem("saveForLater") == null) {
2    ...localStorage.setItem("saveForLater", JSON.stringify([]))
3  }
4  if (localStorage.getItem("removedWish") == null) {
5    ...localStorage.setItem("removedWish", JSON.stringify([]));
6  }
7
8  let total_main_price = 0;
9  let discount_price = 0;
10 let total_price = 0;
11
12 function showcartItems() {
13   ...showremovedWisher();
14   ...showSaveForLater();
15   ...let showCartItems = document.getElementById("showCartItems");
16   ...showCartItems.innerHTML = null;
17   ...let cartItem = JSON.parse(localStorage.getItem("cartItem"));
18   ...for (let i = 0; i < cartItem.length; i++) {
19     ...let div = document.createElement("div");
20     ...div.innerHTML = `<img src=${cartItem[i].img_source}>
21     ...<div>
22     ...<h3>${cartItem[i].course_name}</h3>
23     ...<p>${cartItem[i].Author}</p>
24     ...</div>
25     ...<div>
26     ...<button onclick="removeCourseFromCart(${i})">remove</button>
27     ...<button onclick="saveForLater(${i})">Save for later</button>
28     ...<button onclick="moveToWishList(${i})">Move to wishlist</button>
29     ...</div>
30     ...</div>
31     ...<h3>${cartItem[i].price}</h3>
32     ...<p>${cartItem[i].mainPrice}</p>
33     ...</div>`;
34     ...div.style.display = "flex";
35     ...showCartItems.appendChild(div);
36   ...}
37   ...let cartItem = JSON.parse(localStorage.getItem("cartItem"));
38   ...console.log(cartItem.length)
39   ...let noOfItemsInshop = document.getElementById("noOfItemsInshop");
40   ...noOfItemsInshop.innerHTML = `<div>${cartItem.length} Courses are in cart</div>`;

```

Рисунок 3.16 – Програмна реалізація Cart

- збереження стану кошика. Модуль Cart здатний зберігати стан кошика між сеансами користувача. Це означає, що якщо користувач вийде з сайту та повернеться пізніше, його кошик залишиться незмінним, що дозволяє

продовжити покупки без необхідності додавати курси знову як показано на (рис. 3.17).

Недоліки використання модуля Cart:

- ризик втрати даних. Якщо користувач не авторизується на сайті або не створить обліковий запис перед додаванням товарів до кошика, існує ризик втрати даних, оскільки стан кошика залежить від сеансу користувача. Це може призвести до втрати вибраних товарів, якщо користувач очистить кеш, видалить файли cookie або використовує різні пристрої для доступу до сайту;
- обмеження функціональності без авторизації. Деякі функції модуля Cart, такі як збереження стану кошика між сеансами або доступ до історії замовлень, можуть бути обмежені для незареєстрованих користувачів. Це може обмежити функціональність і зручність використання кошика для таких користувачів;
- технічні проблеми. Модуль Cart може стикатися з технічними проблемами, такими як збої в роботі, помилки обробки замовлень або проблеми зі збереженням даних. Це може призвести до незавершеного оформлення замовлення або втрати даних кошика. Необхідно приділяти увагу тестуванню та вдосконаленню модуля Cart для забезпечення його надійності та безперебійної роботи.

The screenshot displays the Udemy checkout interface. At the top, the 'Checkout' section includes a 'Billing Address' form with dropdowns for 'Country' (set to 'Ukraine') and 'State/Union Territory' (set to 'Kyiv'). Below this, four payment methods are listed: 'Credit or Debit Card' (selected), 'Pay with Net Banking', 'Pay with UPI', and 'Pay with PayTm'. A card details form is visible with fields for card number, expiration date (03/2022), and CVV (511). A 'Remember this Card' checkbox is checked. To the right, the 'Summary' section shows a 'Total Price' of \$9100, a 'Discount Price' of -\$8190, and a final 'Total' of \$910. A red 'Complete Payment' button is present. Below the summary, a green 'Payment Successful' message is displayed. At the bottom, the 'Order Details' section is partially visible.

Рисунок 3.17 – Оформлення покупок ІТ– курсів

Враховуючи переваги та недоліки, модуль Cart є важливою складовою веб-магазину, що дозволяє користувачам зручно вибирати, керувати та оформляти замовлення на курси програмування. Забезпечення правильної реалізації та функціональності модуля Cart є важливим завданням при розробці кваліфікаційної роботи для створення задоволення та зручності для користувачів вашого веб-магазину.

3.5.4 Модуль AfterLogin

Модуль AfterLogin наведено на рисунку 3.18, в кваліфікаційній відіграє важливу роль у післяавторизаційному процесі та надає додаткові можливості користувачам після успішного входу в систему. Описавши переваги та недоліки використання цього модулю, можна виділити наступні основні переваги та аспекти.

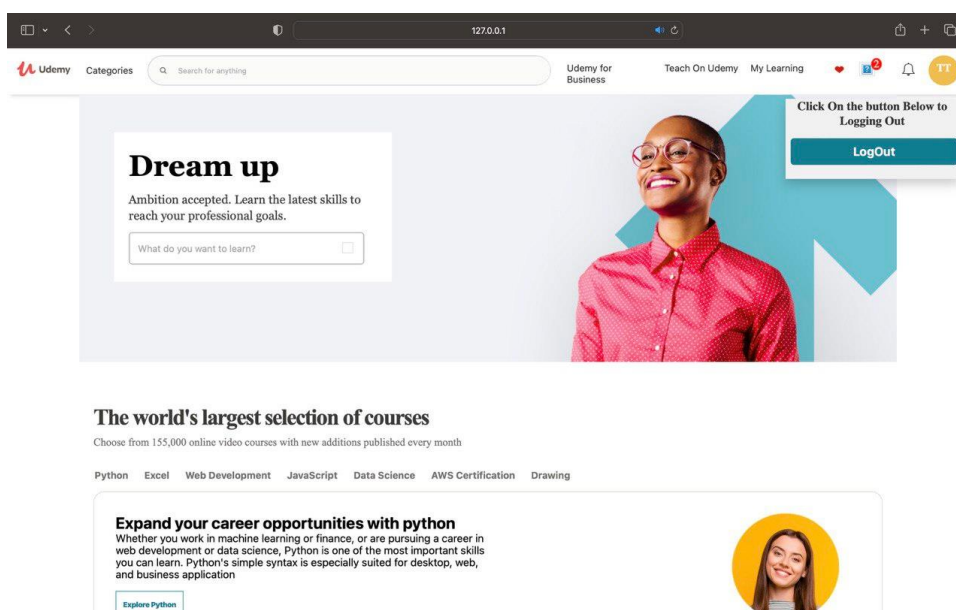


Рисунок 3.18 – Приклад модулю AfterLogin

Переваги використання модуля AfterLogin:

- персоналізоване вітання. Модуль AfterLogin дозволяє вітати користувача особистим повідомленням або інформаційним сповіщенням після успішного входу в систему. Це створює враження індивідуального підходу та забезпечує більш особистий досвід для користувача;

- доступ до особистого кабінету. Модуль AfterLogin надає можливість користувачам зручно отримувати доступ до свого особистого кабінету, де вони можуть переглядати свої дані, історію замовлень, внесені оплати та іншу важливу інформацію. Це сприяє зручності та самостійності користувачів у взаємодії з системою;
- редагування профілю. Модуль AfterLogin дозволяє користувачам змінювати свої особисті дані, такі як ім'я, контактна інформація, пароль тощо. Це надає можливість користувачам оновлювати та налаштовувати свої дані в зручний для них спосіб;
- закладки та облікові записи. Модуль AfterLogin дозволяє користувачам зберігати курси або сторінки, які їм сподобалися, у закладках або облікових записах для подальшого перегляду або вибору. Це сприяє зручності та швидкості вибору курсів та полегшує навігацію по веб-магазину.

Недоліки використання модулю AfterLogin:

- ризик безпеки. Модуль AfterLogin пов'язаний з обробкою та зберіганням особистих даних користувачів. Це може належним чином збільшити вимоги до безпеки системи, включаючи захист від несанкціонованого доступу та зловживань. Необхідно ретельно розробити та впровадити механізми захисту даних для запобігання можливим загрозам безпеки;
- складність реалізації. Модуль AfterLogin вимагає належного розроблення та інтеграції з іншими складовими системи, такими як база даних користувачів та інші модулі. Це може вимагати додаткових зусиль та досліджень для забезпечення коректної роботи модулю;
- підтримка та супровід. Після впровадження модулю AfterLogin, необхідно забезпечити його підтримку та супровід, включаючи регулярні оновлення, виправлення помилок та розширення функціональності. Це важливо для забезпечення безперебійної роботи та задоволення користувачів;
- адаптація до змін. В разі змін вимог або функціональності системи, модуль AfterLogin може потребувати додаткових змін та адаптацій. Необхідно

забезпечити гнучкість та легкість розширення модулю для впровадження майбутніх змін у системі.

Враховуючи переваги та недоліки, модуль AfterLogin є важливою складовою веб-магазину, що дозволяє користувачам отримати більш персоналізований та зручний досвід після входу в систему. Його правильна реалізація та функціональність визначатимуть задоволення та задачі користувачів при взаємодії з веб-магазином курсів програмування.

3.5.5 Реалізація інших компонентів

Udemy–copy:

– модуль «udemy–copy» (рис. 3.19) реалізований за допомогою мови розмітки HTML та CSS і складається з наступних елементів.

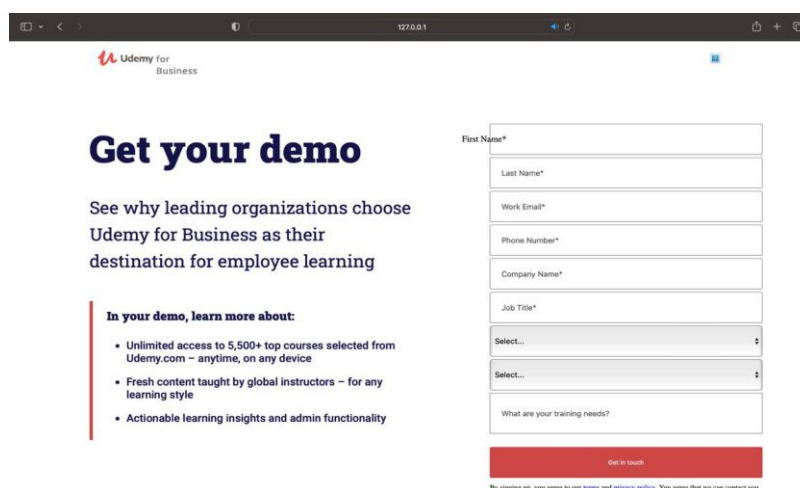


Рисунок 3.19 – Приклад реалізації компоненти

Зовнішній вигляд і стилізація:

- у файлі CSS («udemyCopy.css») задаються стилі для всіх елементів модулю, включаючи шрифти, кольори, розміри і положення;
- використовуються зовнішні шрифти з Google Fonts, такі як «Open Sans», «Roboto Slab», «Roboto»;
- логотип «Udemy» відображається з допомогою тега та має клас «headImg» як показано на рисунку 3.20;

– інші логотипи, такі як «Booking.com», «Apple», «Survey Monkey», «Eventbrite», «Adidas» та «PayPal», також відображаються з допомогою тега `<img>` і мають відповідні класи для стилізації.

Розмітка модулю:

- заголовок сторінки (`<title>`) містить текст «Document»;
- модуль розбитий на декілька блоків (`<div>`) з відповідними класами для стилізації;
- верхній блок (`<header>`) містить логотип «Udemy», текст «for Business» та кнопку з хрестиком;
- блоки з текстом та формою для заповнення даних користувача знаходяться всередині контейнерів `<div>` з класами «box» та «box1»;
- в розділі «Get your demo» відображаються заголовок, опис та список переваг використання «Udemy for Business»;
- форма заповнення даних користувача містить поля для введення інформації, такі як ім'я, прізвище, електронна пошта, номер телефону, назва компанії та посада;
- у формі також є випадаючі списки (`<select>`) для вибору кількості співробітників та розміру компанії, а також поле вводу для опису потреб у навчанні;
- кнопка «Get in touch» служить для надсилання заповненої форми.

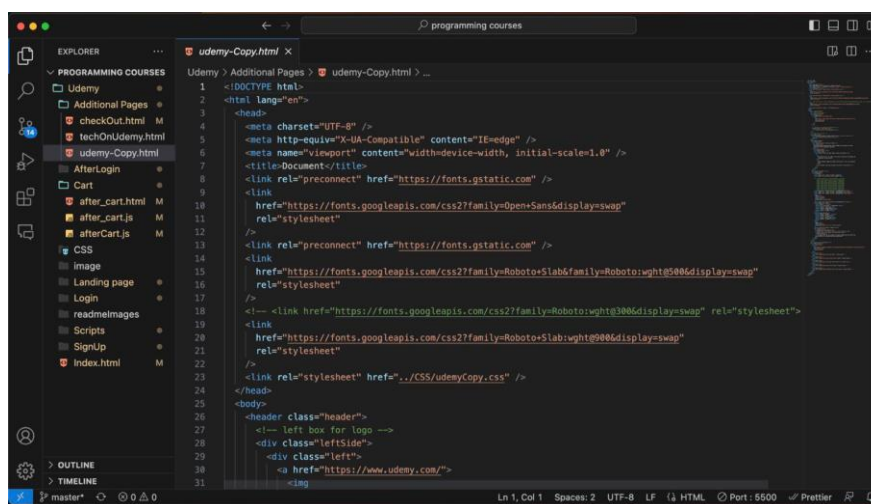


Рисунок 3.20 – Програмна реалізація

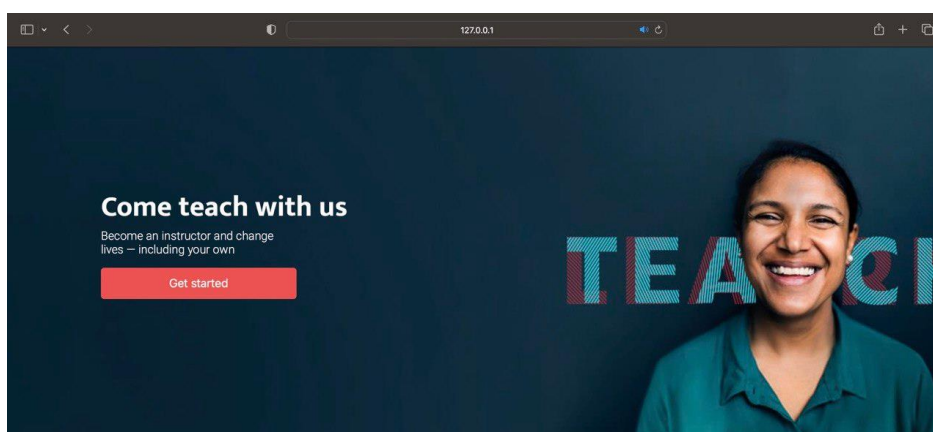
Взаємодія та переходи:

- логотип «Udemy» та текст «for Business» перенаправляють користувача на головну сторінку «<https://www.udemy.com/>»;
- кнопка з хрестиком перенаправляє користувача на сторінку «<https://business.udemy.com>»;
- у формі є посилання на «terms» та «privacy policy», які ведуть на відповідні сторінки.

Цей модуль реалізовує інтерфейс для отримання демонстраційного доступу до «Udemy for Business». Він має зручну розмітку та стилізацію, а також включає форму для заповнення основних даних користувача. Після заповнення форми, користувач може надіслати її за допомогою кнопки «Get in touch».

techOnUdemy.

Модуль techOnUdemy представлено на рисунку 3.21 на веб-сторінці використовується для презентації та привертання уваги користувачів до можливостей викладання на платформі Udemy. Основною метою цього модуля є переконання відвідувачів сторінки стати інструктором на Udemy та надати їм інформацію про переваги цієї платформи.



So many reasons to start

Рисунок 3.21 – Реалізація techOnUdemy

- зображення та заголовки. Банер з привабливим зображенням та заголовком «Come teach with us» (Приєднуйтеся до нас як викладач) привертає увагу користувачів;
- короткий опис. Внутрішній блок містить короткий опис як наведено на рисунку 3.22, який заохочує відвідувачів стати інструктором та змінити життя, включаючи своє власне;

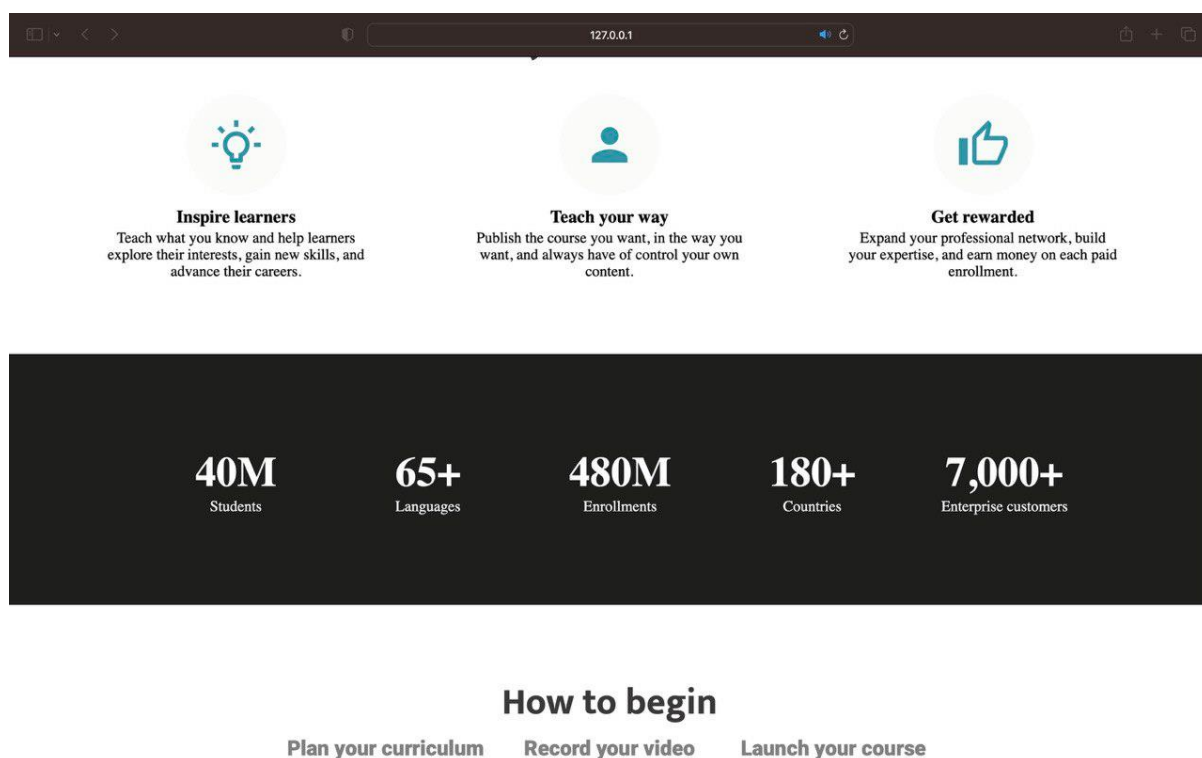
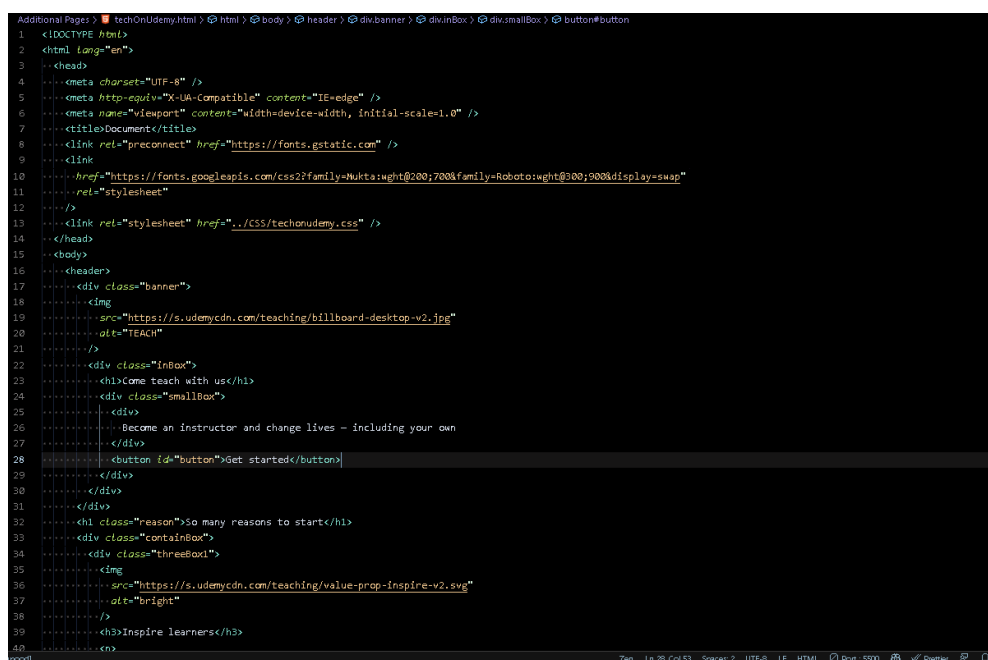


Рисунок 3.22 – Продовження сторінки

Модуль містить наступні елементи для досягнення своєї мети:

- три причини. Секція «So many reasons to start» (Багато причин розпочати) вказує на три ключові переваги викладання на Udeacity: надихати навчання, мати контроль над власним вмістом та отримувати винагороду за кожну оплату;
- статистика. Секція «blackCont» відображає різні статистичні дані, що свідчать про успіх та масштабність платформи, такі як кількість студентів, кількість мов, кількість записів, кількість країн та кількість корпоративних клієнтів;

- кроки для початку. Розділ «How to begin» (Як почати) надає короткий огляд процесу розпочатку викладання на Udemy;
- кнопки. Секція «buttons» містить три кнопки, які представляють кроки у процесі створення та запуску курсу на платформі, включаючи планування навчального плану, запис відео та запуск курсу показано на рисунку 3.23.



```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8" />
5 <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7 <title>Document</title>
8 <link rel="preconnect" href="https://fonts.gstatic.com" />
9 <link
10 href="https://fonts.googleapis.com/css2?family=Nunito:wght@200;700&family=Roboto:wght@300;900&display=swap"
11 rel="stylesheet"
12 />
13 <link rel="stylesheet" href="css/techonudemy.css" />
14 </head>
15 <body>
16 <div class="banner">
17 
21 </div>
22 <div class="inbox">
23 <h1>Come teach with us</h1>
24 <div class="smallBox">
25 <div>
26 Become an instructor and change lives – including your own
27 </div>
28 <button id="button">Get started</button>
29 </div>
30 </div>
31 </div>
32 <div class="reason">So many reasons to start</h1>
33 <div class="containBox">
34 <div class="threeBox1">
35 
39 <h3>Inspire learners</h3>
40 </div>

```

Рисунок 3.23 – Програмна реалізація

Цей модуль допомагає залучити увагу відвідувачів сторінки та надати їм інформацію про переваги викладання на Udemy, спонукаючи їх розпочати свою власну кар'єру як викладачів на цій платформі.

Висновки до розділу 3

В третьому розділі кваліфікаційної роботи бакалавра на тему «Розробка веб-магазину з продажу курсів програмування» представлені ключові переваги, які надає веб-магазин студентам та викладачам.

Широкий вибір курсів: Розроблений веб-магазин надає широкий вибір курсів з програмування. Це дозволяє студентам обирати навчальні програми, що

відповідають їхнім інтересам та потребам. Великий асортимент курсів сприяє підвищенню можливостей для навчання і розвитку учнів.

Зручність та доступність: Розроблений веб-магазин надає зручний і легкий спосіб замовлення та оплати курсів програмування. Студентам не потрібно відвідувати фізичний магазин або надсилати запити електронною поштою. Вони можуть легко здійснити покупку відразу через веб-сайт, що забезпечує швидкий та безпечний процес придбання.

Взаємодія з викладачами: Веб-магазин забезпечує можливість взаємодії студентів з викладачами курсів. Це дозволяє студентам отримувати підтримку, задавати питання та отримувати консультації з тем, які їх цікавлять. Взаємодія з викладачами сприяє більш глибокому засвоєнню матеріалу та можливості вирішення проблем учнів.

В цілому, третій розділ кваліфікаційної роботи підкреслює важливість розробки веб-магазину для продажу курсів програмування, оскільки він надає студентам широкі можливості для навчання та розвитку. Веб-магазин забезпечує зручний процес придбання та сприяє взаємодії між студентами та викладачами, що сприяє покращенню якості навчання.

ВИСНОВКИ

В кваліфікаційній роботі бакалавра на тему «Розробка веб-магазину з продажу курсів програмування» була проведена детальна розробка та реалізація веб-магазину, спрямованого на продаж курсів програмування. Робота включала аналіз вимог, проектування архітектури, розробку функціональності та дизайну, а також тестування рішення.

Згідно з поставленим завданням данної кваліфікаційної роботи бакалавра, було створено веб-сервіс, у якому був створений веб-сервіс, який надає користувачам можливість переглядати, замовляти та навчатися курсам програмування. Веб-сервіс містить каталог курсів, де користувачі можуть докладно ознайомитися з кожним курсом, його характеристиками, включаючи назву, опис, автора, тривалість та ціну.

Було розроблено можливість додавати ці курси до кошика та здійснювати замовлення. Під час оформлення замовлення користувачі вводять необхідні контактні дані, такі як ім'я та адресу електронної пошти, щоб забезпечити зв'язок і доставку. Після підтвердження замовлення, користувачі отримують доступ до куплених курсів та можуть розпочати навчання.

Було створено у веб-сервісі функцію авторизації та реєстрації користувачів. Користувачі можуть створювати свої облікові записи та авторизуватися на сайті. Це дозволяє зберігати їх замовлення, налаштування та інші особисті дані у безпечному особистому кабінеті.

Було додано можливість додавати нові курси програмування до каталогу. Він може заповнити інформацію про курс, включаючи назву, опис, автора, тривалість та ціну, а також завантажити зображення, щоб візуально представити товар.

В результаті, розроблений веб-сервіс надає зручний інтерфейс для користувачів, де вони можуть переглядати, замовляти та навчатися курсам програмування. Користувачі можуть авторизуватися та мати свій ос.

Отже, у процесі розробки веб-сервісу були враховані принципи веб-дизайну та забезпечення захисту даних користувачів. Також була здійснена інтеграція з платіжними системами для забезпечення онлайн-платежів та оплати за курси програмування.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Udemy // Udemy.com. URL: <https://www.udemy.com/> (дата звернення 20.03.2023)
2. Coursera // Coursera.org. URL: <https://www.coursera.org/> (дата звернення 13.04.2023)
3. Codecademy // Codecademy.com. URL: <https://www.codecademy.com/> (дата звернення 17.04.2023)
4. Nodejs // Nodejs.org. URL: <https://nodejs.org/uk> (дата звернення 25.04.2023)
5. Duckett J. JavaScript and jQuery: Interactive Front-End Web Development Jon Duckett. Web Site Development URL: <https://www.wiley.com/en-us/Javascript+and+jQuery:+Interactive+Front+End+Web+Developmentp/9781118531648/> (дата звернення 12.04.2023).
6. JavaScript Tutoria W3Schools URL: <https://www.w3schools.com/js/> (дата звернення 2.05.2023).
7. JavaScript: Mozilla Developer Network (MDN) URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (дата звернення 13.05.2023).
8. A Modern Introduction to Programming. Eloquent JavaScript URL: <https://eloquentjavascript.net/> (дата звернення 25.05.2023).
9. Stack Overflow URL: <https://stackoverflow.com> (дата звернення 9.05.2023).
10. CSS-Tricks URL: <https://css-tricks.com> (дата звернення 7.05.2023).
11. A List Apart URL: <https://alistapart.com/> (дата звернення 13.05.2023).
12. CSS Grid Garden URL: <https://cssgridgarden.com> (дата звернення 25.05.2023).
13. JavaScript.info URL: <https://javascript.info> (дата звернення 06.05.2023).
14. DevDocs URL: <https://devdocs.io/> (дата звернення 14.05.2023).

ДОДАТКИ

Додаток А

Лістинг 1 – відображення першого підменю

```
function showfirst(e) {
    showOverlay();
    let subfirst2 = document.getElementById("subfirst2");
    let closebtn = document.getElementById("closebtn");
    let first = document.getElementById("first");
    closebtn.style.display = "block";
    first.style.display = "block";
    subfirst2.innerHTML = null;
    for (let i = 0; i < arrFirst.length; i++) {
        let div = document.createElement("div");
        div.innerHTML = `\${arrFirst\[i\]\[1\]}</a><button
onclick="showSecond\(\${i}\)">&#8250;</button>`;
        subfirst2.appendChild\(div\);
    }
}
```

Кінець лістингу 1

Лістинг 2 – Обробка Onclick подій

```
window.onclick = function (event) {
    console.log(event.target);
    var closebtn = document.getElementById("closebtn");
    var dropdown1 = document.getElementById("first");
    var dropdown2 = document.getElementById("second");

    if (
        event.target.matches("#overlay") ||
        event.target.matches("SVG") ||
        event.target.matches("path")
    ) {
        closebtn.style.display = "none";
        dropdown1.style.display = "none";
        dropdown2.style.display = "none";
        overlay.style.display = "none";
    }

    var black_overlay = document.getElementById("black_overlay");
    var close_black_overlay = document.getElementById("close_black_overlay");
    if (event.target.matches("#close_black_overlay") ||
        event.target.matches("#black_overlay")) {
        black_overlay.style.display = "none";
    }
};
```

Кінець лістингу 2

Лістинг 3 – Функція вмісту курсу Python

```

function showPython() {
    slideCourses[0].innerHTML = null;
    var pythonSlider = document.createElement("div");
    pythonSlider.setAttribute("class", 'pythonSlider');
    var upperContentSlider = document.createElement("div")
    upperContentSlider.setAttribute("class", "upperContentSlider");
    var upperContentSlider_div1 = document.createElement("div");
    var upperContentSlider_div1_h1 = document.createElement("h1");
    upperContentSlider_div1_h1.innerHTML = "Expand your career opportunities with
python";
    var upperContentSlider_div1_p1 = document.createElement("p");
    upperContentSlider_div1_p1.innerHTML = "Whether you work in machine learning or
finance, or are pursuing a career in web development or data science, Python is
one of the most important skills you can learn. Python's simple syntax is
especially suited for desktop, web, and business application"
    var upperContentSlider_div1_button1 = document.createElement("button");
    upperContentSlider_div1_button1.innerHTML = "Explore Python";
    upperContentSlider_div1.append(upperContentSlider_div1_h1,
upperContentSlider_div1_p1, upperContentSlider_div1_button1);
    upperContentSlider.appendChild(upperContentSlider_div1);
    var upperContentSlider_div2 = document.createElement("div");
    var upperContentSlider_div1_img1 = document.createElement("img");
    upperContentSlider_div1_img1.src = "https://s.udemycdn.com/topic-images/lohp-
topic-banners/GettyImages-1166389425_opt.jpg";
    upperContentSlider_div1_img1.alt = "womenLogo"
    upperContentSlider.appendChild(upperContentSlider_div2);
    upperContentSlider_div2.appendChild(upperContentSlider_div1_img1)
    pythonSlider.appendChild(upperContentSlider)
    var main = document.createElement("div");
    main.setAttribute("id", "main");
    var main_button1 = document.createElement("button");
    main_button1.setAttribute("class", "previous round");
    main_button1.innerHTML = "&#8249;"
    main_button1.onclick = function () {
        plusSlides(-1);
    };
    main.appendChild(main_button1);
    var main_div1 = document.createElement("div");
    main_div1.setAttribute("class", "slides fade");
    main_div1.onmouseover = function () {
        popupAddToCart(0);
    };
    var main_div2 = document.createElement("div");
    main_div2.setAttribute("class", "slides fade");
    main_div2.onmouseover = function () {
        popupAddToCart(1);
    };
    var main_div3 = document.createElement("div");
    main_div3.setAttribute("class", "slides fade");

```

```

main_div3.onmouseover = function () {
    popupAddToCart(2);
};
var main_div4 = document.createElement("div");
main_div4.setAttribute("class", "slides fade");
main_div4.onmouseover = function () {
    popupAddToCart(3);
};
var main_div5 = document.createElement("div");
main_div5.setAttribute("class", "slides fade");
main_div5.onmouseover = function () {
    popupAddToCart(4);
};
main.append(main_div1, main_div2, main_div3, main_div4, main_div5);

var main_button2 = document.createElement("button");
main_button2.setAttribute("class", "next round");
main_button2.innerHTML = "&#8250;";
main_button2.onclick = function () {
    plusSlides(1);
};
var pythonSlider_div_3 = document.createElement("div");
pythonSlider_div_3.setAttribute("class", "popup-div-addtocart")
var pythonSlider_div_4 = document.createElement("div");
pythonSlider_div_4.setAttribute("class", "big-popup-before-addtocart")
main.appendChild(main_button2);
pythonSlider.appendChild(main);
pythonSlider.appendChild(pythonSlider_div_3, pythonSlider_div_4);
slideCourses[0].appendChild(pythonSlider);

```

Кінець лістингу 3

Лістинг 4 – Функції додавання курсів та приховання вмісту

```

function popupAddToCart(i) {
    let indexOfContent = i + slideIndex;
    let popupAddToCart = document.getElementsByClassName("popup-div-addtocart")[0];
    popupAddToCart.style.display = "inline-block";
    popupAddToCart.innerHTML =
`<h3>${educationContent[indexOfContent].course_name}</h3>
    <p><span style="background-color:#ffe799;padding:3px;border-radius:5px;font-weight:600"> BestSeller </span><span style="margin-left:15px;color:green;">Updated
<strong>${educationContent[indexOfContent].updatedAt}</strong></span></p>
    <span>${educationContent[indexOfContent].totalHoursOfVideo} total
hours <ul><li>Beginners Level</li><li>Subtitles</li></ul></span>
    <p>${educationContent[indexOfContent].courseHeading}</p>
    <ul>

```

```

        <li>Learn to use Python professionally, learning both Python 2
and Python 3!</li>
        <li>Create games with Python, like Tic Tac Toe and
Blackjack!</li>
        <li>Learn advanced Python features, like the collections module
and how to work with timestamps!</li>
    </ul>
    <a href="#black_overlay"onclick="showbigpopup(${indexOfContent})"
>Add to cart</a><span class="wishlistButton" onclick="addToWishlist()"><i
class="far fa-heart fa-2x" style="color: #107d90;"></i></span>`;

    popupAddToCart.style.top = "-280px";
    if (i < 3) {
        popupAddToCart.style.left = (i + 1) * 20 + "%";
    } else if (i == 3) {
        popupAddToCart.style.left = "27%";
    } else {
        popupAddToCart.style.left = "47%";
    }
}
function disablepopupAddToCart() {
    document.getElementsByClassName("popup-div-addtocart")[0].style = "none";
}
}
window.onload = showPython();
function showWebdevelopment() {
    var slideCourses = document.getElementsByClassName("slideCourses");
    slideCourses[0].innerHTML = null;
    var pythonSlider = document.createElement("div");
    pythonSlider.setAttribute("class", 'pythonSlider');
    var upperContentSlider = document.createElement("div")
    upperContentSlider.setAttribute("class", "upperContentSlider");
    var upperContentSlider_div1 = document.createElement("div");
    var upperContentSlider_div1_h1 = document.createElement("h1");
    upperContentSlider_div1_h1.innerText = "Build websites and applications with
Web Development";
    var upperContentSlider_div1_p1 = document.createElement("p");
    upperContentSlider_div1_p1.innerText = "Whether you work in machine learning or
finance, or are pursuing a career in web development or data science, Python is one
of the most important skills you can learn. Python's simple syntax is especially
suited for desktop, web, and business application"
    var upperContentSlider_div1_button1 = document.createElement("button");
    upperContentSlider_div1_button1.innerText = "Explore Webdevelopment";
    upperContentSlider_div1.append(upperContentSlider_div1_h1,
upperContentSlider_div1_p1, upperContentSlider_div1_button1);
    upperContentSlider.appendChild(upperContentSlider_div1);
    var upperContentSlider_div2 = document.createElement("div");
    var upperContentSlider_div1_img1 = document.createElement("img");
    upperContentSlider_div1_img1.src = "https://s.udemycdn.com/topic-images/lohp-
topic-banners/GettyImages-1175266254_opt.jpg";

```

```

upperContentSlider_div1_img1.alt = "manLogo"
upperContentSlider.appendChild(upperContentSlider_div2);
upperContentSlider_div2.appendChild(upperContentSlider_div1_img1)
pythonSlider.appendChild(upperContentSlider)
var main = document.createElement("div");
main.setAttribute("id", "main");
var main_button1 = document.createElement("button");
main_button1.innerHTML = "&#8249;"
main_button1.setAttribute("class", "previous round");
main_button1.onclick = function () {
    plusSlides(-1);
};
main.appendChild(main_button1);
var main_div1 = document.createElement("div");
main_div1.setAttribute("class", "slides fade");
main_div1.onmouseover = function () {
    popupAddToCart(0);
};
var main_div2 = document.createElement("div");
main_div2.setAttribute("class", "slides fade");
main_div2.onmouseover = function () {
    popupAddToCart(1);
};
var main_div3 = document.createElement("div");
main_div3.setAttribute("class", "slides fade");
main_div3.onmouseover = function () {
    popupAddToCart(2);
};
var main_div4 = document.createElement("div");
main_div4.setAttribute("class", "slides fade");
main_div4.onmouseover = function () {
    popupAddToCart(3);
};
var main_div5 = document.createElement("div");
main_div5.setAttribute("class", "slides fade");
main_div5.onmouseover = function () {
    popupAddToCart(4);
};
main.append(main_div1, main_div2, main_div3, main_div4, main_div5);

var main_button2 = document.createElement("button");
main_button2.setAttribute("class", "next round");
main_button2.innerHTML = "&#8250;"
main_button2.onclick = function () {
    plusSlides(1);
};
var pythonSlider_div_3 = document.createElement("div");
pythonSlider_div_3.setAttribute("class", "popup-div-addtocart")
var pythonSlider_div_4 = document.createElement("div");
pythonSlider_div_4.setAttribute("class", "big-popup-before-addtocart")

```

```

main.appendChild(main_button2);
pythonSlider.appendChild(main);
pythonSlider.append(pythonSlider_div_3, pythonSlider_div_4);
slideCourses[0].appendChild(pythonSlider);

```

Кінець лістингу 4

Лістинг 5 – Створення впливаючих вікон

```

function popupAddToCart(i) {
    let indexOfContent = i + slideIndex;
    let popupAddToCart = document.getElementsByClassName("popup-div-
addtocart")[0];
    popupAddToCart.style.display = "inline-block";
    popupAddToCart.innerHTML =
`<h3>${educationContent[indexOfContent].course_name}</h3>
    <p><span style="background-color:#ffe799;padding:3px;border-
radius:5px;font-weight:600"> BestSeller </span><span style="margin-
left:15px;color:green;">Updated
<strong>${educationContent[indexOfContent].updatedAt}</strong></span></p>
    <span>${educationContent[indexOfContent].totalHoursOfVideo} total
hours <ul><li>Beginners Level</li><li>Subtitles</li></ul></span>
    <p>${educationContent[indexOfContent].courseHeading}</p>
    <ul>
    <li>Learn to use Python professionally, learning both Python 2
and Python 3!</li>
    <li>Create games with Python, like Tic Tac Toe and
Blackjack!</li>
    <li>Learn advanced Python features, like the collections module
and how to work with timestamps!</li>
    </ul>
    <a href="#black_overlay" onclick="showbigpopup(${indexOfContent})"
>Add to cart</a><span class="wishlistButton" onclick="addToWishlist()"><i
class="far fa-heart fa-2x" style="color: #107d90;"></i></span>`;

    popupAddToCart.style.top = "-280px";
    if (i < 3) {
        popupAddToCart.style.left = (i + 1) * 20 + "%";
    } else if (i == 3) {
        popupAddToCart.style.left = "27%";
    } else {
        popupAddToCart.style.left = "47%";
    }
}
function disablepopupAddToCart() {
    document.getElementsByClassName("popup-div-addtocart")[0].style = "none";
}
}

```

Кінець лістингу 5

Лістинг 6 – функція студентській слайдер

```

var studentSlides = document.getElementsByClassName("slides-2");
var noOfDivs = studentSlides.length;
// var a1 = document.getElementsByClassName("previous");
var studentSlideIndex = 0;
showStudSlides(studentSlideIndex);
function plusSlideStud(n) {
    studentSlideIndex += noOfDivs * n - 1;
    if (studentSlideIndex > 1) {
        showStudSlides(studentSlideIndex);
    } else {
        studentSlideIndex = 0;
        showStudSlides(studentSlideIndex);
    }
}

function showStudSlides(a) {
    // for (let i = 0; i < numberOfdivs; i++) {
    //     slides[i].innerHTML = "";
    // }
    if (a + noOfDivs - 1 > studentContent.length) {
        studentSlideIndex = 0;
        a = 0;
    }
    let divNo = 0;
    for (let i = a; i < a + noOfDivs; i++) {
        studentSlides[divNo].style.display = "block";
        studentSlides[
            divNo
        ].innerHTML = `<img      src=${studentContent[i].img_source}><div
class="description"><h3>${studentContent[i].course_name}</h3><p>${studentContent[i]
.Author}</p><div      class="star"><span>${studentContent[i].rating}</span><p
class="Stars"                                style="--rating:
${studentContent[i].rating};"></p><span>&#40;${studentContent[i].people}&#41;</span
></div><div class="slide-price"><h3>&#36  ${studentContent[i].price}</h3><span>&#36
${studentContent[i].mainPrice}</span></div><span
                                id="student-view-
bestseller">Bestseller</span></div>`;
        divNo++;
    }
}

```

Кінець лістингу 6