

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Аналіз інструментаріїв та можливих підходів для створення веб-сайту
для представлення портфоліо

Виконав: студент IV курсу, групи СП-41
спеціальності 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

Гладиный А. О.

(підпис)

(прізвище та ініціали)

Керівник

Бойко І. В.

(підпис)

(прізвище та ініціали)

Нормоконтроль

(підпис)

(прізвище та ініціали)

Завідувач кафедри

(підпис)

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

Тернопіль
2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис) (прізвище та ініціали)
«__» _____ 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

Студенту Гладинюку Антону Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Аналіз інструментаріїв та можливих підходів для створення веб-сайту для представлення портфоліо

Керівник роботи _____
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «__» лютого 2024 року № 4/7-13X

2. Термін подання студентом завершеної роботи _____

3. Вихідні дані до роботи _____

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. РОЗДІЛ 1. Постановка задачі розробки веб-сайту для представлення портфоліо 1.1

Аналіз предметної області, 1.2 Формування вимог розробки веб-портфоліо, 1.3 Пошук

актантів та варіантів використання веб-портфоліо, 1.3.1 Пошук актантів та варіантів

використання веб-портфоліо, 1.3.2 Варіанти використання веб-портфоліо", 1.4 Опис

ключових варіантів використання веб-портфоліо, 1.5 Вибір середовища розробки веб-

портфоліо", 1.5.1 Мова програмування, 1.5.2 База даних,

1.6 Обґрунтування використовуваних технологій розробки веб-портфоліо", 1.7 Висновок до

першого розділу, 2. Проєктування та реалізація веб-сайту для представлення портфоліо",

2.1 Клієнт-серверна архітектура веб-портфоліо", 2.2 Вибір основних залежностей для

розробки веб-сайту для представлення портфоліо", 2.3 Файлова структура веб-портфоліо",

2.4 Структура та особливості використаної БД для розробки веб-портфоліо 2.5

Адміністрування веб-сайту 2.6 Розгортання веб-сайту портфоліо на хостингових

платформах,

2.7 Тестування та використання веб-сайту 2.7.1 Тестування веб-сайту

2.7.2 Використання веб-портфоліо 2.8 Висновки до другого розділу, 3. Безпека

життєдіяльності, основи хорони праці, 3.1 Психологічні причини нещасних випадків і

травматизму, 3.2 Вимоги до профілактичних медичних оглядів для працівників ПК, 3.3

Висновок до третього розділу, Висновки, Перелік використаних джерел, Додатки.

5. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи хорони праці			

6. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи		
2.	Пошук джерел щодо розробки та створення веб-сайтів		
3.	Опрацювання обраних джерел		
4.	Аналіз області розробки веб-сайту, підбір технологій для розробки		
5.	Розробка веб-сайту		
6.	Оформлення розділу «Постановка задачі розробки веб-сайту»		
7.	Оформлення розділу «Проектування та реалізація веб-сайту»		
8.	Виконання завдання до підрозділу «Безпека життєдіяльності»		
9.	Виконання завдання до підрозділу «Основи хорони праці»		
10.	Оформлення кваліфікаційної роботи		
11.	Нормоконтроль		
12.	Перевірка на плагіат		
13.	Попередній захист кваліфікаційної роботи		
14.	Захист кваліфікаційної роботи		

Студент

(підпис)

Гладинюк А. О.

(прізвище та ініціали)

Керівник роботи

(підпис)

Бойко І.В.

(прізвище та ініціали)

АНОТАЦІЯ

Аналіз інструментаріїв та можливих підходів для створення веб-сайту для представлення портфоліо засобами PHP, , HTML та CSS // Кваліфікаційна робота освітнього рівня «Бакалавр» // Gladynyuk Anton Oleksandrovych // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СП-41 // Тернопіль, 2024 // С. 77, рис. – 15, табл. – 1, кресл. – 0 , додат. – 4, бібліогр. – 12.

Ключові слова: php, js, sql, веб-сайт, merch, web, portfolio, developer.

Кваліфікаційна робота присвячена розробці веб-сайту для представлення портфоліо засобами PHP, JavaScript, HTML та CSS.

Метою даної роботи є розробка веб-сайту портфоліо для представлення послуг та виконаних проектів.

В першому розділі кваліфікаційної роботи проведено аналіз предметної області, сформовано основні вимоги для розробки, вибравши при цьому необхідні інструменти, описано взаємодію акторів з системою.

В другому розділі кваліфікаційної роботи розглянуто архітектуру розроблюваного веб-сайту, описано основні залежності та структуру, розглянуто розгортання на хостингових платформах, проведено тестування.

В третьому розділі, було розглянуто психологічні причини нещасних випадків та травматизму та вимоги до медични оглядів працівників за ПК

ANNOTATION

Analysis of tools and possible approaches for creating a website for presenting a portfolio using PHP, , HTML and CSS // Qualification work of the educational level "Bachelor" // Gladynyk Anton Oleksandrovich // Ivan Pulyuy Ternopil National Technical University, Faculty of Computer and Information Sciences of systems and software engineering, department of computer sciences, SP-41 group // Ternopil, 2024 // P. 77, fig. – 15, tab. – 1, armchair. - 0, add. – 4, bibliography - 12.

Keywords: php, js, sql, веб-сайт, merch, web, portfolio, developer.

The qualification work is devoted to the development of the portfolio website using PHP, JavaScript, HTML and CSS.

The purpose of this work is to develop a portfolio website for presenting services and completed projects.

In the first section of the qualification work, the analysis of the subject area was carried out, the main requirements for development were formed, while choosing the necessary tools, the interaction of actors with the system was described.

In the second section of the qualification work, the architecture of the developed website was considered, the main dependencies and structure were described, the deployment on hosting platforms was considered, and testing was carried out.

In the third section, the psychological causes of accidents and injuries and the requirements for medical examinations of employees according to PC were considered

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

HTML – Мова розмітки для створення структури веб-сторінок. HTML визначає, як повинні виглядати елементи сторінки, такі як заголовки, абзаци, посилання і зображення.

CSS – Технологія для визначення стилів веб-сторінок. CSS використовується для встановлення зовнішнього вигляду сторінки, включаючи шрифти, кольори, відступи і макет.

JS – Мова програмування для додавання інтерактивності на веб-сторінки. JS дозволяє створювати анімації, обробляти події користувача і маніпулювати веб-сторінками в реальному часі.

PHP – Скриптова мова для створення динамічних веб-сторінок. PHP обробляє запити сервером, дозволяючи створювати веб-сторінки, які змінюються залежно від даних користувача або бази даних.

SQL – Мова для управління та запитів до баз даних. SQL дозволяє створювати, оновлювати і запитувати дані з бази даних, що є основою для зберігання інформації на веб-сайтах.

FTP – Протокол для передачі файлів між комп'ютерами. FTP дозволяє завантажувати, завантажувати і управляти файлами на сервері через мережу.

API – Інтерфейс, який дозволяє програмам взаємодіяти між собою. API визначає, як програми можуть обмінюватися даними або функціоналом, що дозволяє інтеграцію з іншими системами або сервісами.

VPS – Віртуальний сервер, що працює як окремий фізичний сервер. VPS надає користувачам власне середовище з виділеними ресурсами на фізичному сервері, забезпечуючи кращу продуктивність і контроль.

SSL – Технологія для забезпечення безпечного з'єднання між браузером і сервером. SSL шифрує дані, що передаються між клієнтом і сервером, забезпечуючи захист від перехоплення і підробки.

JSO – Об'єкт JavaScript для організації даних. JSO використовується для зберігання і обробки інформації в структурованому вигляді, що дозволяє легше маніпулювати даними в JavaScript.

JSON – Формат обміну даними, легкий для читання людиною. JSON використовується для передачі структурованих даних між сервером і клієнтом, що дозволяє легко обробляти і використовувати інформацію.

UI – Частина веб-сайту, з якою взаємодіє користувач. UI включає всі елементи, що відображаються на сторінці, такі як кнопки, меню, форми та інші інтерактивні компоненти.

UX – Загальний досвід користувача при використанні продукту. UX охоплює всі аспекти взаємодії користувача з продуктом, забезпечуючи зручність, ефективність і задоволення від використання.

DB – Організована структура для зберігання даних. DB забезпечує систематичне зберігання і управління інформацією, що дозволяє швидко отримувати потрібні дані.

SEO – Оптимізація для підвищення видимості сайту у пошукових системах. SEO включає різні стратегії і техніки для покращення рейтингу сайту в результатах пошукових запитів.

HTML5 – Остання версія мови розмітки HTML. HTML5 додає нові елементи і атрибути, що покращують можливості для створення мультимедійного контенту і інтерактивних веб-додатків.

CSS3 – Остання версія CSS з додатковими можливостями. CSS3 включає нові властивості і функції, що дозволяють створювати більш складні і ефективні стилі для веб-сторінок.

HTTP – Протокол для передачі даних між клієнтом і сервером. HTTP визначає правила обміну інформацією в Інтернеті, забезпечуючи запити клієнта і відповіді сервера.

ЗМІСТ

АНОТАЦІЯ	4
ANNOTATION	5
ЗМІСТ	9
ВСТУП.....	11
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ ВЕБ-САЙТУ ДЛЯ ПРЕДСТАВЛЕННЯ ВЕБ-ПОРТФОЛІО	12
1.1 Аналіз предметної області	12
1.2 Формування вимог розробки веб-портфоліо	15
1.3 Пошук актантів та варіантів використання веб-портфоліо	17
1.3.1 Варіанти використання веб-портфоліо.....	19
1.4 Опис ключових варіантів використання веб-портфоліо.....	21
1.5 Вибір середовища розробки веб-портфоліо.....	23
1.5.1 Мова програмування	25
1.5.2 База даних.....	26
1.6 Обґрунтування використовуваних технологій розробки веб-портфоліо	29
1.7 Висновок до першого розділу	31
2. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВЕБ-САЙТУ ДЛЯ ПРЕДСТАВЛЕННЯ ПОРТФОЛІО	32
2.1 Клієнт-серверна архітектура веб-портфоліо.....	32
2.2 Вибір основних залежностей для розробки веб-сайту для представлення портфоліо.....	34
2.3 Файлова структура веб-портфоліо	36
2.4 Структура та особливості використаної БД для розробки веб-портфоліо	39
2.5 Адміністрування веб-сайту.....	41
2.6 Розгортання веб-сайту портфоліо на хостингових платформах.....	43
2.7 Тестування та використання веб-сайту	46
2.7.1 Тестування веб-сайту	46

2.7.2	Використання веб-портфоліо	48
2.8	Висновки до другого розділу	51
3.	БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ.....	53
3.1	Психологічні причини нещасних випадків і травматизму	53
3.2	Вимоги до профілактичних медичних оглядів для працівників ПК	56
3.3	Висновок до третього розділу	59
	ВИСНОВКИ	62
	ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	64
	Додаток А	67
	Лістинг 1 – Код файлу «index.php».....	67
	Додаток Б.....	71
	Лістинг 2 – Код файлу «admin.js».....	71
	Додаток В	78
	Лістинг 3 – Код файлу «script-index.js».....	78
	Додаток Г	79
	Лістинг 4 – Код файлу «script-payment.js».....	79

ВСТУП

Актуальність теми. У сучасному світі цифрових технологій та інтернету, наявність персонального веб-сайту стала важливим аспектом професійного розвитку для багатьох спеціалістів, особливо розробників. Веб-сайт портфоліо є візитною карткою, яка дозволяє демонструвати свої проекти, навички та досягнення потенційним роботодавцям та клієнтам. Крім того, веб-сайт може бути ефективним інструментом для розвитку особистого бренду та підвищення видимості в професійному співтоваристві.

Зростаюча популярність дистанційного навчання та онлайн-курсів ще більше підкреслює важливість створення платформи, яка пропонує навчальні матеріали. Це створило великий попит на якісний освітній контент у сфері розробки. Веб-сайт, який пропонує курси з розробки, може задовольнити цей попит та надати цінні ресурси для навчання широкому колу користувачів.

Розробка індивідуальних веб-сайтів для бізнесів і персональних проектів також залишається актуальною. Багато малих та середніх підприємств шукають професійних розробників для створення веб-сайтів, які відповідали б їхнім потребам та сприяли бізнесу. Веб-сайт, що пропонує послуги з розробки сайтів, може стати ефективною платформою для залучення таких клієнтів, забезпечуючи їх високоякісними та персоналізованими рішеннями.

Мета і задачі дослідження. Метою даного проекту є створення інтерактивного веб-сайту портфоліо розробника за допомогою PHP та JavaScript, який буде виконувати дві основні функції: презентація портфоліо розробника та надання можливості придбання курсів з розробки та замовлення послуг з розробки веб-сайтів. Цей проект покликаний не лише продемонструвати навички розробника, але й створити зручну платформу для навчання та надання професійних послуг, що відповідають сучасним потребам ринку.

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ ВЕБ-САЙТУ ДЛЯ ПРЕДСТАВЛЕННЯ ВЕБ-ПОРТФОЛІО

1.1 Аналіз предметної області

Аналіз предметної області є ключовим етапом у розробці веб-сайту, що дозволяє глибоко зрозуміти потреби користувачів, визначити основні вимоги до функціональності системи та оцінити конкурентне середовище. Основна мета цього розділу – зібрати та проаналізувати інформацію, яка допоможе створити веб-сайт, що відповідатиме очікуванням користувачів і вирізнятиметься серед інших подібних платформ.

Розробка персонального веб-сайту для розробника, який об'єднує функції портфоліо, платформи для продажу курсів та майданчика для замовлення веб-розробок, потребує детального вивчення кількох ключових аспектів. Перш за все, важливо розуміти, хто є цільовою аудиторією цього веб-сайту. Основними користувачами можуть бути потенційні клієнти, які шукають кваліфікованого розробника для створення власних веб-сайтів, а також студенти та фахівці, які прагнуть покращити свої навички у сфері веб-розробки. Розуміння потреб цих груп дозволить розробити інтерфейс та функціональність, що максимально задовольняють їх очікування.

Далі, важливо провести аналіз конкурентного середовища. На ринку існує чимало веб-сайтів розробників, які пропонують схожі послуги. Для того щоб створити конкурентоспроможний продукт, необхідно дослідити сильні та слабкі сторони конкурентів. Це дозволить виявити найкращі практики та уникнути помилок, що були зроблені іншими. Особливу увагу слід звернути на дизайн, юзабіліті, функціональність та унікальні особливості веб-сайтів конкурентів.

Технічний аналіз є не менш важливим аспектом. Він включає вивчення сучасних технологій та інструментів, які можуть бути використані для розробки веб-сайту. У даному випадку, вибір зупинено на PHP та JavaScript як

основних мовах програмування. PHP забезпечує надійність та гнучкість серверної частини, тоді як JavaScript додає динамічності та інтерактивності клієнтській частині. Вибір цих технологій обґрунтований їхньою популярністю, великою кількістю бібліотек та інструментів, а також активною підтримкою спільноти розробників.

При розробці веб-сайту портфоліо, що включає функції продажу курсів та надання послуг зі створення веб-сайтів, важливо розглянути різні технології, щоб вибрати найкращий варіант для реалізації проекту. Незважаючи на те, що проект вже використовує PHP і JavaScript, є інші технології, які могли б бути корисними в цьому контексті.

Однією з таких альтернатив є використання мови програмування Python разом з фреймворком Django. Python відомий своєю простотою та читабельністю коду, що значно спрощує процес розробки. Django, в свою чергу, є потужним фреймворком, який дозволяє швидко створювати веб-додатки, забезпечуючи високу продуктивність завдяки вбудованим інструментам для адміністрування, аутентифікації користувачів, обробки форм та управління базами даних. Завдяки своїй структурі, Django забезпечує високий рівень безпеки, що є важливим аспектом для будь-якого веб-додатку, особливо якщо мова йде про обробку фінансових транзакцій та особистої інформації користувачів.

Іншою перспективною технологією є Ruby on Rails, яка також дозволяє швидко створювати веб-додатки завдяки принципу "конвенція над конфігурацією". Ruby on Rails надає розробникам набір стандартних налаштувань, що дозволяє зосередитися на написанні бізнес-логіки замість конфігураційного коду. Крім того, Rails має багату екосистему бібліотек (гемів), які можуть бути легко інтегровані в проект для додавання нових функцій. Хоча продуктивність Ruby може бути нижчою в порівнянні з іншими мовами, зручність і швидкість розробки часто компенсують цей недолік.

Node.js разом з Express.js також є популярним вибором для розробки веб-додатків. Використання JavaScript як на стороні клієнта, так і на сервері дозволяє створювати більш узгоджений кодовий базис та спрощує навчання для розробників. Node.js відомий своєю високою продуктивністю завдяки асинхронній обробці запитів, що робить його ідеальним для додатків з високим навантаженням. Express.js, в свою чергу, забезпечує мінімалістичний фреймворк для побудови серверної логіки, дозволяючи створювати масштабовані додатки з мінімальними затратами.

Ще однією альтернативою є Java з використанням Spring Boot. Java є однією з найбільш популярних мов програмування, особливо у корпоративному середовищі, завдяки своїй масштабованості та надійності. Spring Boot дозволяє швидко створювати готові до використання додатки з мінімальними налаштуваннями. Висока продуктивність Java, разом з можливістю легкого розширення функціональності, робить Spring Boot привабливим вибором для великих та складних проектів. Хоча налаштування та конфігурація можуть бути складнішими для новачків, результати часто виправдовують зусилля, особливо коли мова йде про стабільні та безпечні додатки.

Розгляд різних технологій дозволяє знайти найкраще рішення для конкретного проекту, враховуючи всі вимоги та обмеження. Кожна з розглянутих технологій має свої переваги та недоліки, і вибір залежить від специфіки проекту, досвіду розробника та ресурсів, доступних для реалізації.

Також важливо врахувати аспекти безпеки та захисту даних. Веб-сайт, що обробляє фінансові транзакції та зберігає особисту інформацію користувачів, повинен мати надійні механізми захисту від кібератак, витоку даних та інших загроз. Це включає використання SSL-сертифікатів, надійних методів шифрування, а також регулярне оновлення програмного забезпечення.

Розуміння вимог до контенту є ще одним важливим аспектом аналізу предметної області. Веб-сайт повинен мати зрозумілу та структуровану

інформацію про надані послуги, доступні курси, портфоліо проектів та контактні дані. Контент повинен бути легко доступним і зрозумілим для користувачів, що сприятиме позитивному користувацькому досвіду.

На основі проведеного аналізу предметної області можна сформулювати чіткі вимоги до веб-сайту, що дозволить створити конкурентоспроможний, функціональний та безпечний продукт, який відповідатиме потребам користувачів та допоможе розробнику ефективно презентувати свої навички, продавати курси та надавати послуги з розробки веб-сайтів.

1.2 Формування вимог розробки веб-портфоліо

Формування вимог до розробки веб-сайту є критичним етапом, який визначає функціональність, зручність використання та ефективність кінцевого продукту. Вимоги повинні бути чітко визначені та задокументовані, щоб уникнути непорозумінь між замовником і розробником, а також забезпечити, що всі очікування будуть виконані.

Перш за все, необхідно визначити основні функціональні вимоги, які включають всі ключові функції та можливості, що мають бути реалізовані на веб-сайті. Основна мета проекту – створити інтерактивний веб-сайт портфоліо розробника, який також дозволяє продавати курси з розробки та надавати послуги зі створення веб-сайтів. Це означає, що веб-сайт повинен мати такі функціональні можливості, як відображення портфоліо проектів, функціонал для покупки курсів, система для замовлення послуг, а також засоби для реєстрації та авторизації користувачів.

Однією з ключових вимог є розробка інтерфейсу для демонстрації портфоліо проектів. Цей інтерфейс повинен бути візуально привабливим та інтуїтивно зрозумілим, щоб потенційні клієнти могли легко ознайомитися з

роботами розробника. Портфоліо має містити детальний опис кожного проекту, включаючи технічні характеристики, використані технології та результати. Важливо також передбачити можливість додавання нових проектів та редагування існуючих через адміністраторську панель.

Іншою важливою вимогою є розробка функціоналу для продажу курсів. Це включає створення сторінок з описом курсів, їх програмою, відгуками користувачів та вартістю. Система повинна забезпечувати можливість безпечної оплати за допомогою різних платіжних систем, а також надавати доступ до матеріалів курсу після успішної оплати. Крім того, необхідно передбачити функціонал для управління курсами, включаючи додавання нових курсів, оновлення інформації та видалення застарілих.

Система замовлення послуг з розробки веб-сайтів також є критичною частиною проекту. Вона повинна включати форму для запиту на розробку, де клієнти можуть вказувати свої вимоги та побажання. Форма має бути простою та зрозумілою, забезпечуючи зручний спосіб для клієнтів зв'язатися з розробником. Також важливо передбачити механізм для обробки цих запитів, їх збереження в базі даних та можливість швидкої відповіді клієнтам.

Нефункціональні вимоги також грають важливу роль у розробці веб-сайту. До них належать вимоги до продуктивності, безпеки, зручності використання та масштабованості. Веб-сайт повинен мати високу продуктивність, забезпечуючи швидке завантаження сторінок та оперативну обробку запитів. Безпека є критичним аспектом, особливо при обробці фінансових транзакцій та особистих даних користувачів. Необхідно передбачити механізми захисту від різних типів атак, таких як SQL-ін'єкції, CSRF та XSS. Зручність використання включає створення інтуїтивно зрозумілого інтерфейсу, який забезпечить позитивний досвід користувачів. Масштабованість означає, що веб-сайт повинен бути готовий до збільшення навантаження та розширення функціональності без значних змін у архітектурі.

На завершення, необхідно врахувати вимоги до сумісності та підтримки різних пристроїв і браузерів. Веб-сайт повинен коректно відображатися та працювати як на десктопах, так і на мобільних пристроях. Це включає адаптивний дизайн та тестування на різних платформах, щоб забезпечити однаковий користувацький досвід незалежно від використовуваного пристрою.

Формування вимог до розробки веб-сайту є важливим кроком, який закладає основу для успішної реалізації проекту. Чітке визначення функціональних та нефункціональних вимог дозволяє створити веб-сайт, який буде відповідати очікуванням користувачів, забезпечить високу продуктивність та безпеку, а також надасть можливість для подальшого розвитку та масштабування.

1.3 Пошук актантів та варіантів використання веб-портфоліо

Для успішної розробки веб-сайту портфоліо, який включає функції продажу курсів та надання послуг з розробки веб-сайтів, необхідно провести аналіз актантів, або зацікавлених сторін, а також варіантів використання веб-сайту. Актанти – це всі учасники, які взаємодіють з веб-сайтом, а варіанти використання – це способи, якими ці учасники використовуватимуть функціональні можливості сайту.

Основними акторами веб-сайту є потенційні клієнти, студенти, які прагнуть придбати курси, та сам розробник, який адмініструватиме сайт. Потенційні клієнти – це люди або організації, які шукають професійні послуги зі створення веб-сайтів. Вони зацікавлені у перегляді портфоліо розробника, щоб оцінити його навички та досвід, а також у можливості легко зв'язатися з розробником для обговорення деталей проекту. Студенти, які прагнуть

придбати курси, шукають якісні навчальні матеріали з веб-розробки, доступ до яких вони можуть отримати після оплати. Розробник є ключовим адміністратором сайту, відповідальним за додавання нових проектів у портфоліо, управління курсами та обробку запитів на розробку веб-сайтів.

Потенційні клієнти взаємодіють із веб-сайтом, переглядаючи портфоліо, заповнюючи форми замовлення послуг і здійснюючи оплату. Вони заходять на сайт, щоб ознайомитися з роботами розробника, вивчити деталі виконаних проектів та переконатися у високій якості послуг. Клієнти також використовують контактні форми або систему повідомлень для зв'язку з розробником, щоб обговорити свої потреби та отримати пропозиції щодо реалізації своїх проектів.

Студенти використовують веб-сайт для пошуку та придбання курсів з веб-розробки. Вони переглядають опис курсів, вивчають програму навчання та відгуки інших студентів, щоб прийняти рішення про покупку. Після вибору курсу студенти здійснюють оплату через захищену платіжну систему та отримують доступ до навчальних матеріалів. Важливо, щоб процес покупки був простим і зручним, а також щоб студенти мали можливість швидко отримати допомогу у разі виникнення питань або проблем.

Розробник, як адміністратор сайту, має широкий спектр функцій і завдань. Він додає нові проекти до портфоліо, оновлює інформацію про існуючі проекти та видаляє застарілі або неактуальні дані. Крім того, розробник створює і управляє курсами, додаючи нові навчальні матеріали, оновлюючи програми курсів та відповідаючи на запитання студентів. Він також обробляє запити клієнтів на розробку веб-сайтів, відповідає на повідомлення та координує процес співпраці з клієнтами. Розробник також відповідає за підтримку безпеки веб-сайту, забезпечення його безперебійної роботи та проведення регулярних оновлень.

Кожен з актантів має свої специфічні варіанти використання веб-сайту, які необхідно враховувати при розробці функціональності та інтерфейсу.

Потенційні клієнти потребують зручного доступу до портфоліо та контактних форм, студенти – інтуїтивно зрозумілого інтерфейсу для пошуку та придбання курсів, а розробник – потужних інструментів для адміністрування сайту. Врахування потреб кожної з цих груп користувачів дозволить створити зручний та ефективний веб-сайт, який задовольнить всі їхні очікування та вимоги.

1.3.1 Варіанти використання веб-портфоліо

Для успішної розробки веб-сайту портфоліо, який включає функції продажу курсів та надання послуг з розробки веб-сайтів, необхідно провести аналіз актантів, або зацікавлених сторін, а також варіантів використання веб-сайту. Актанти – це всі учасники, які взаємодіють з веб-сайтом, а варіанти використання – це способи, якими ці учасники використовуватимуть функціональні можливості сайту.

Основними акторами веб-сайту є потенційні клієнти, студенти, які прагнуть придбати курси, та сам розробник, який адмініструватиме сайт. Потенційні клієнти – це люди або організації, які шукають професійні послуги зі створення веб-сайтів. Вони зацікавлені у перегляді портфоліо розробника, щоб оцінити його навички та досвід, а також у можливості легко зв'язатися з розробником для обговорення деталей проекту. Студенти, які прагнуть придбати курси, шукають якісні навчальні матеріали з веб-розробки, доступ до яких вони можуть отримати після оплати. Розробник є ключовим адміністратором сайту, відповідальним за додавання нових проектів у портфоліо, управління курсами та обробку запитів на розробку веб-сайтів.

Потенційні клієнти взаємодіють із веб-сайтом, переглядаючи портфоліо, заповнюючи форми замовлення послуг і здійснюючи оплату. Вони заходять

на сайт, щоб ознайомитися з роботами розробника, вивчити деталі виконаних проектів та переконатися у високій якості послуг.

Студенти використовують веб-сайт для пошуку та придбання курсів з веб-розробки. Вони переглядають опис курсів, вивчають програму навчання та відгуки інших студентів, щоб прийняти рішення про покупку. Після вибору курсу студенти здійснюють оплату через захищену платіжну систему та отримують доступ до навчальних матеріалів. Важливо, щоб процес покупки був простим і зручним, а також щоб студенти мали можливість швидко отримати допомогу у разі виникнення питань або проблем.

Розробник, як адміністратор сайту, має широкий спектр функцій і завдань. Він додає нові проекти до портфоліо, оновлює інформацію про існуючі проекти та видаляє застарілі або неактуальні дані. Крім того, розробник створює і управляє курсами, додаючи нові навчальні матеріали, оновлюючи програми курсів та відповідаючи на запитання студентів. Він також обробляє запити клієнтів на розробку веб-сайтів, відповідає на повідомлення та координує процес співпраці з клієнтами.

Кожен з актантів має свої специфічні варіанти використання веб-сайту, які необхідно враховувати при розробці функціональності та інтерфейсу. Потенційні клієнти потребують зручного доступу до портфоліо та контактних форм, студенти – інтуїтивно зрозумілого інтерфейсу для пошуку та придбання курсів, а розробник – потужних інструментів для адміністрування сайту.

Короткий опис варіантів використання наведено в таблиці 1.1.

Таблиця 1.1 – Варіантів використання веб-сайту портфоліо

Актор	Найменування	Формулювання
Користувач	Перегляд вмісту, перегляд товарів, замовлення послуг	Доступ до функціонування веб- сайту, формулювання замовлень
Адміністратор	Перегляд, додавання, редагування, видалення	Перегляд вмісту сайту, Додавання контенту, редагування вмісту, видалення, в тому числі й інших користувачів.

1.4 Опис ключових варіантів використання веб-портфоліо

Для повного розуміння функціональності веб-сайту портфоліо розробника важливо детально описати ключові варіанти його використання. Це дозволяє забезпечити, що всі необхідні аспекти взаємодії користувачів з сайтом будуть враховані під час розробки. Варіанти використання можна розділити на декілька основних сценаріїв, які включають взаємодію з різними компонентами веб-сайту.

Одним з основних варіантів використання веб-сайту є перегляд портфоліо проектів. Користувачі, які відвідують сайт, зокрема потенційні

клієнти, зацікавлені в ознайомленні з попередніми роботами розробника. Портфоліо повинно бути організоване таким чином, щоб користувачі могли легко знайти потрібну інформацію. Кожен проект повинен містити опис, використані технології, цілі проекту та досягнуті результати. Крім того, бажано додати візуальні матеріали, такі як скріншоти або відео, які наочно демонструють виконану роботу. Цей варіант використання включає такі підваріанти, як фільтрація проектів за категоріями, пошук по ключових словах та перегляд детальної інформації про кожен проект.

Один з ключових сценаріїв – це покупка курсів з веб-розробки. Користувачі повинні мати можливість переглядати доступні курси, ознайомлюватися з їх описом, програмою та відгуками інших студентів. Після вибору курсу користувачі переходять до оформлення замовлення, де їм пропонується ввести платіжні дані та підтвердити покупку. Після успішної оплати курс повинен стати доступним для користувача, який може переглядати навчальні матеріали, завантажувати файли та виконувати завдання. Важливо забезпечити безпеку платіжних даних користувачів, використовуючи захищені протоколи для передачі інформації та інтеграцію з надійними платіжними системами.

Іншим важливим варіантом використання є замовлення послуг зі створення веб-сайтів. Потенційні клієнти повинні мати можливість легко знайти інформацію про доступні послуги, ознайомитися з умовами співпраці та подати запит на розробку. Цей процес включає заповнення спеціальної форми, де клієнти вказують свої вимоги та побажання, а також залишають контактні дані для зворотного зв'язку. Після подання запиту розробник отримує повідомлення та може зв'язатися з клієнтом для уточнення деталей та обговорення умов співпраці. Важливо, щоб цей процес був максимально прозорим та зрозумілим для клієнтів, що сприятиме побудові довірчих відносин з ними.

1.5 Вибір середовища розробки веб-портфоліо

При розробці веб-сайту портфоліо розробника, який включає функції продажу курсів та надання послуг зі створення веб-сайтів, важливим кроком є вибір середовища розробки. Для цього проекту було обрано Notepad++, що є популярним текстовим редактором серед веб-розробників. Вибір цього середовища обумовлений його численними перевагами, які роблять його ефективним інструментом для розробки без використання складних IDE або фреймворків [3].

Notepad++ є легким, швидким і безкоштовним текстовим редактором, який підтримує широкий спектр мов програмування, включаючи PHP і JavaScript. Його основні переваги включають простоту використання, низький рівень ресурсозатрат і високу продуктивність, що робить його ідеальним для невеликих і середніх проектів, таких як розробка веб-сайту портфоліо

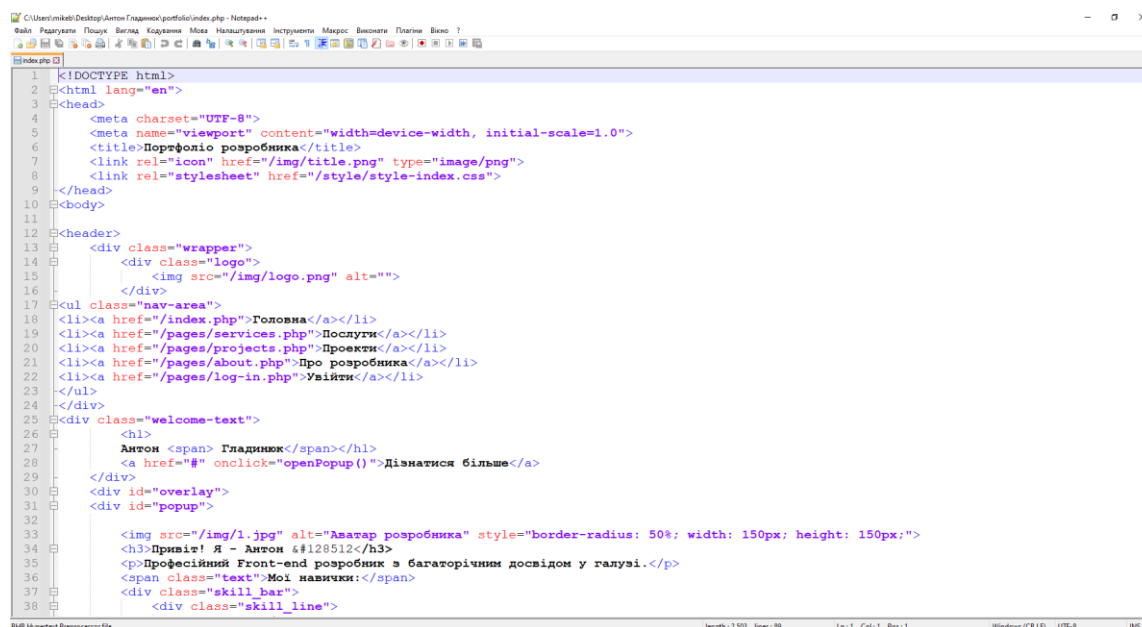


Рисунок 1.1 – Середовище розробки «Notepad++»

Однією з головних причин вибору Notepad++ є його можливості для синтаксичного підсвічування та автозаповнення коду. Ці функції значно прискорюють написання коду, знижуючи ймовірність помилок і полегшуючи читання коду. Крім того, Notepad++ підтримує роботу з різними форматами файлів, що дозволяє зручно редагувати як PHP-файли, так і JavaScript-код, а також HTML і CSS. Це забезпечує єдність і узгодженість коду на всіх етапах розробки.

Notepad++ також має широкий вибір плагінів, які можуть бути легко встановлені для розширення функціональності редактора. Наприклад, плагіни для роботи з версіями контролю, такі як Git, або плагіни для перевірки синтаксису та форматування коду, можуть значно підвищити продуктивність розробника. Це особливо корисно при роботі з великим обсягом коду або при командній розробці, коли необхідно швидко знаходити і виправляти помилки.

Ще однією важливою характеристикою Notepad++ є його можливість роботи з кількома документами одночасно, що дозволяє легко перемикатися між файлами проекту, не закриваючи їх. Це значно спрощує процес розробки, особливо при роботі з великою кількістю файлів, як це часто буває при розробці веб-сайтів.

Враховуючи необхідність тестування коду на різних етапах розробки, Notepad++ дозволяє інтегруватися з різними інструментами для тестування, такими як PHPUnit для PHP або Jasmine для JavaScript. Це забезпечує можливість автоматизованого тестування і швидкої перевірки працездатності коду, що підвищує якість кінцевого продукту.

З урахуванням цих переваг, вибір Notepad++ як середовища розробки для цього проекту здається обґрунтованим рішенням. Його функціональні можливості, простота використання і ефективність роблять його відмінним інструментом для реалізації веб-сайту портфоліо розробника, який відповідає сучасним вимогам до веб-розробки. Крім того, Notepad++ є безкоштовним і доступним інструментом, що робить його ідеальним для розробників, які

працюють самостійно або в невеликих командах, не маючи потреби в дорогих і важких IDE.

1.5.1 Мова програмування

Мова програмування є основним інструментом для створення функціональності веб-сайту портфоліо розробника, що включає продаж курсів та надання послуг зі створення веб-сайтів. Для цього проекту була обрана мова програмування PHP, яка є однією з найпопулярніших мов для розробки веб-додатків. Вибір PHP обумовлений його універсальністю, потужними можливостями для створення динамічного контенту та широкою підтримкою з боку веб-спільноти.

PHP (Hypertext Preprocessor) – це серверна мова програмування, яка спеціалізується на створенні веб-додатків. Вона дозволяє розробникам інтегрувати динамічний контент у веб-сторінки, обробляти запити користувачів, керувати базами даних та виконувати інші серверні функції. PHP добре інтегрується з HTML, що дозволяє легко вставляти PHP-код у HTML-документи, створюючи динамічні веб-сторінки без необхідності використовувати окремі файли для серверної логіки та вмісту.

Однією з ключових переваг PHP є його легкість у вивченні та використанні. Веб-розробники можуть швидко освоїти основи PHP і почати створювати функціональні веб-додатки з мінімальними затратами часу. PHP також має велику кількість бібліотек і фреймворків, таких як Laravel, Symfony та CodeIgniter, які спрощують процес розробки, забезпечуючи готові рішення для багатьох типових завдань, таких як аутентифікація користувачів, обробка форм та управління базами даних.

PHP підтримує роботу з різними базами даних, включаючи MySQL, PostgreSQL та SQLite, що дозволяє легко зберігати і обробляти великі обсяги даних, необхідних для веб-сайту з функціями продажу курсів і замовлення послуг. Інтеграція з базами даних здійснюється за допомогою стандартних функцій PHP або з використанням бібліотек, таких як PDO (PHP Data Objects), що забезпечує безпечну та ефективну роботу з базами даних.

Безпека є ще одним важливим аспектом, який забезпечує PHP. Мова включає численні функції для захисту від типових веб-атак, таких як SQL-ін'єкції, XSS (межовий скриптинг) та CSRF (підробка запитів між сайтами). Використання таких функцій і практик, як підготовлені запити, захист від атак на сесії та обробка входу даних, дозволяє знизити ризики безпеки і забезпечити надійний захист даних користувачів.

PHP також відомий своєю масштабованістю, що робить його підходящим для створення як малих, так і великих веб-додатків. Він може працювати на різних платформах, таких як Windows, Linux та macOS, що забезпечує гнучкість у виборі середовища розробки та хостингу.

Таким чином, PHP є ідеальним вибором для розробки веб-сайту портфоліо розробника, що має функції продажу курсів та надання послуг зі створення веб-сайтів. Його простота у використанні, потужність, можливості для інтеграції з базами даних і високий рівень безпеки роблять його ефективним інструментом для реалізації всіх необхідних функцій проекту.

1.5.2 База даних

Для зберігання та обробки даних веб-сайту портфоліо розробника, який включає продаж курсів та надання послуг зі створення веб-сайтів, важливо

вибрати ефективну систему управління базами даних. Розглядаються декілька популярних варіантів, кожен з яких має свої особливості та переваги.

Одним з варіантів може бути MySQL. MySQL є однією з найбільш популярних систем управління базами даних з відкритим вихідним кодом. Вона відома своєю стабільністю, масштабованістю та високою продуктивністю. MySQL підтримує великі обсяги даних, що робить її ідеальною для веб-додатків з високим навантаженням. Однією з ключових переваг MySQL є її здатність обробляти запити дуже швидко завдяки оптимізованим алгоритмам індексації та кешування.

MySQL добре інтегрується з PHP, що є важливим для цього проекту. PHP-розробники мають широкий доступ до функцій для роботи з MySQL, що робить процес розробки простішим і швидшим. MySQL також підтримує транзакції, що забезпечує цілісність даних і можливість виконання складних запитів до бази даних. Крім того, MySQL має потужні засоби для управління правами доступу, що дозволяє забезпечити безпеку даних користувачів.

Ще одним варіантом може бути PostgreSQL яка є ще однією популярною системою управління базами даних з відкритим вихідним кодом. Вона відома своєю гнучкістю, потужністю та підтримкою складних запитів, включаючи транзакції з ACID-сумісністю, що робить її ідеальною для великих і складних додатків. PostgreSQL підтримує широкий спектр типів даних, що дозволяє зберігати дані в різних форматах, включаючи JSON та XML.

PostgreSQL також має розвинуті механізми для забезпечення цілісності даних і підтримки високої доступності. Проте, на відміну від MySQL, PostgreSQL може вимагати більшої кількості ресурсів для налаштування та експлуатації, що може бути недоцільно для проектів з обмеженими ресурсами або невеликими обсягами даних.

Також може бути SQLite це є дуже простою в налаштуванні та використанні системою управління базами даних, яка зберігає дані у файлі, а не на окремому сервері. Це робить її ідеальною для малих проектів або для

розробки прототипів. SQLite легко інтегрується з PHP і не потребує додаткового налаштування сервера бази даних, що значно спрощує розгортання додатку.

Однак, SQLite має обмеження щодо масштабованості та продуктивності, що робить її непридатною для великих або інтенсивних додатків, таких як веб-сайт портфоліо з функціями продажу курсів і замовлення послуг.

Після розгляду різних варіантів було вирішено використовувати PHPMyAdmin з MySQL для бази даних цього проекту. PHPMyAdmin є інтерфейсом для адміністрування баз даних MySQL через веб-браузер, що забезпечує зручний і інтуїтивно зрозумілий спосіб керування базою даних. Його основні переваги включають простоту у використанні, зручний інтерфейс для створення, редагування та управління таблицями, а також можливість виконання SQL-запитів безпосередньо через веб-інтерфейс

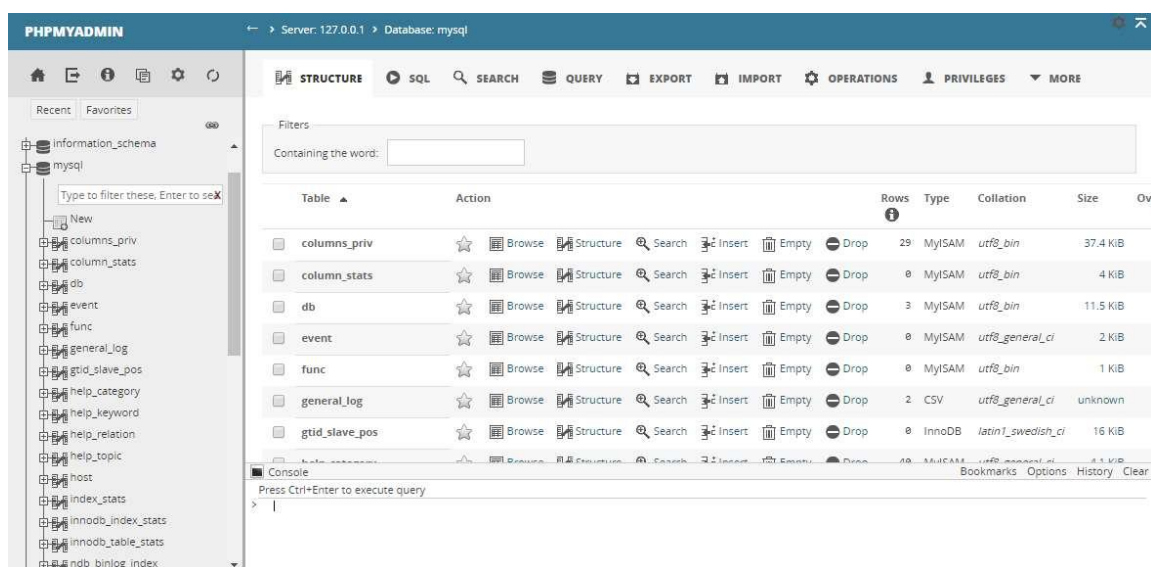


Рисунок 1.2 – Інтерфейс «PHPMyAdmin»

MySQL, у свою чергу, забезпечує необхідну продуктивність, масштабованість і безпеку для проекту. Її підтримка великих обсягів даних, високоякісне виконання запитів та інтеграція з PHP роблять її ідеальним вибором для цього веб-сайту. Застосування PHPMyAdmin дозволяє легко

управляти базою даних, виконувати регулярні завдання адміністрування і забезпечувати швидкий доступ до даних, що значно спрощує процес розробки та підтримки проекту.

1.6 Обґрунтування використовуваних технологій розробки веб-портфоліо

У цьому проекті було обрано технології, які забезпечують максимальну ефективність, зручність у використанні та відповідність сучасним вимогам до веб-розробки. Обґрунтування вибору кожної з них ґрунтується на їхніх перевагах та здатності задовольнити специфічні потреби проекту.

PHP було обрано як основну мову програмування для розробки серверної логіки веб-сайту. Цей вибір обумовлений кількома ключовими факторами. По-перше, PHP є однією з найпопулярніших мов для розробки веб-додатків, яка добре інтегрується з HTML, що дозволяє створювати динамічні веб-сторінки без складних конструкцій. По-друге, PHP підтримує роботу з різними базами даних, включаючи MySQL, що робить його ідеальним для проекту з необхідністю обробки великих обсягів даних. По-третє, PHP має широку спільноту та велике число бібліотек і фреймворків, що значно спрощує розробку і тестування коду.

MySQL було обрано як систему управління базами даних завдяки її стабільності, масштабованості та високій продуктивності. MySQL добре підтримує великий обсяг даних, що особливо важливо для цього проекту, де необхідно зберігати інформацію про курси, замовлення клієнтів, дані користувачів та інші важливі дані. Інтеграція MySQL з PHP забезпечує ефективне виконання SQL-запитів і забезпечує цілісність даних через

підтримку транзакцій. Крім того, MySQL має добре розвинену систему управління правами доступу, що допомагає забезпечити безпеку даних.

Для створення фронтенду веб-сайту обрано HTML, CSS та JavaScript. HTML використовується для створення структури веб-сторінок, забезпечуючи правильне відображення контенту в браузері. CSS додає стилі, які роблять веб-сайт привабливим і зручним для користувачів, дозволяючи налаштовувати вигляд сторінок відповідно до вимог дизайну. JavaScript додає динамічність, дозволяючи реалізовувати взаємодію з користувачем, анімації та інтерактивні елементи. Використання цих технологій забезпечує високу якість і доступність веб-сторінок на різних пристроях.

Notepad++ було обрано як середовище розробки через його простоту, легкість у використанні і потужні можливості для роботи з PHP і JavaScript. Notepad++ забезпечує ефективне синтаксичне підсвічування, автозаповнення коду та підтримку роботи з кількома документами одночасно, що значно покращує продуктивність розробки. Його легкість і швидкість дозволяють розробникам зосередитися на написанні коду без зайвих затримок, що є важливим для динамічної розробки проекту.

РНРMyAdmin було обрано для управління базою даних MySQL через його зручний веб-інтерфейс. Цей інструмент дозволяє легко створювати, редагувати та управляти таблицями бази даних, виконувати SQL-запити і здійснювати адміністративні функції без необхідності глибоких знань SQL. РНРMyAdmin значно спрощує процес адміністрування бази даних, що є важливим для забезпечення ефективного управління даними веб-сайту.

1.7 Висновок до першого розділу

У результаті аналізу та обґрунтування вибору технологій для розробки веб-сайту портфоліо розробника, можна зробити кілька важливих висновків. Перш за все, ключовими аспектами, які визначили вибір технологій, є ефективність, зручність у використанні та відповідність сучасним стандартам веб-розробки.

PHP було обрано як основну мову програмування завдяки його простоті, потужності та гнучкості у інтеграції з різними базами даних. Це дозволяє легко обробляти динамічний контент, виконувати складні запити до бази даних і забезпечувати високу продуктивність веб-додатку. Вибір MySQL як системи управління базами даних підкреслює необхідність надійного зберігання і обробки великого обсягу даних, а також забезпечення їх цілісності та безпеки.

HTML, CSS та JavaScript забезпечують високоякісний фронтенд, що дозволяє створювати привабливі та інтерактивні веб-сторінки, які відповідають сучасним вимогам до дизайну і функціональності. Використання Notepad++ як середовища розробки забезпечує зручність і продуктивність при написанні коду, а PHPMyAdmin спрощує управління базою даних, що значно підвищує ефективність адміністрування.

Таким чином, вибрані технології забезпечують комплексний підхід до розробки веб-сайту, який відповідає всім необхідним вимогам проекту. Вони гарантують не тільки високу якість і надійність кінцевого продукту, але й зручність у використанні для розробників, що значно спрощує процес створення та підтримки веб-сайту. Розробка з використанням цих технологій дозволить створити функціональний, безпечний і привабливий веб-сайт, який відповідатиме всім очікуванням користувачів і бізнес-цілям проекту.

2. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВЕБ-САЙТУ ДЛЯ ПРЕДСТАВЛЕННЯ ПОРТФОЛІО

2.1 Клієнт-серверна архітектура веб-портфоліо

Клієнт-серверна архітектура є основою для побудови веб-сайту портфоліо розробника, який включає функції продажу курсів та надання послуг зі створення веб-сайтів. Ця архітектура забезпечує розподіл обчислювальних завдань між клієнтом і сервером, що дозволяє ефективно використовувати ресурси обох сторін і забезпечує швидкий доступ до інформації для кінцевих користувачів.

Клієнт-серверна архітектура веб-портфоліо складається з двох основних компонентів: клієнтської частини (фронтенду) та серверної (бекенду)

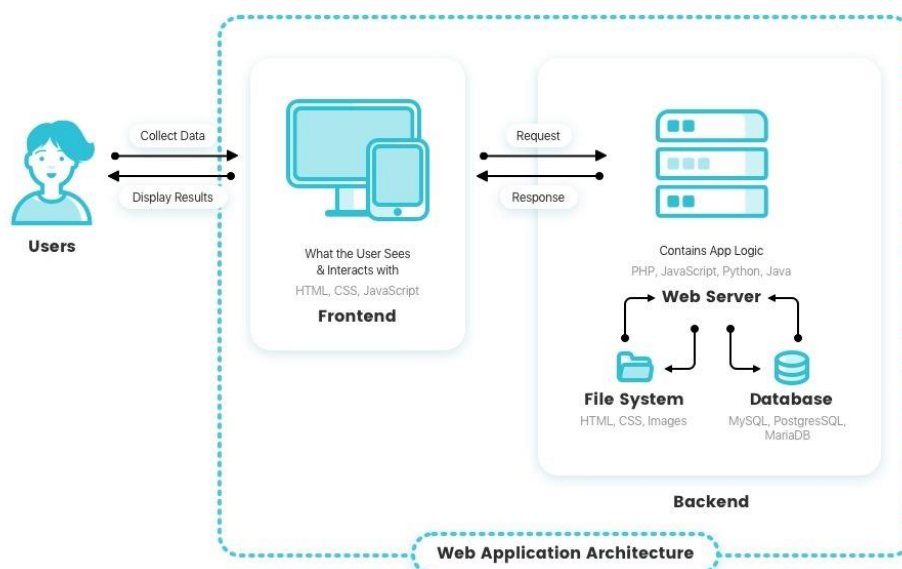


Рисунок 2.1 – Клієнт-серверна архітектура веб-сайту

Клієнтська частина відповідає за відображення даних і взаємодію з користувачем. Вона розроблена за допомогою HTML, CSS і JavaScript. HTML забезпечує структуру веб-сторінок, CSS відповідає за стилізацію та візуальне оформлення, а JavaScript додає динамічність та інтерактивність, що дозволяє

користувачам здійснювати різноманітні дії на веб-сайті, такі як перегляд курсів, оформлення замовлень та взаємодія з формами [1, 2].

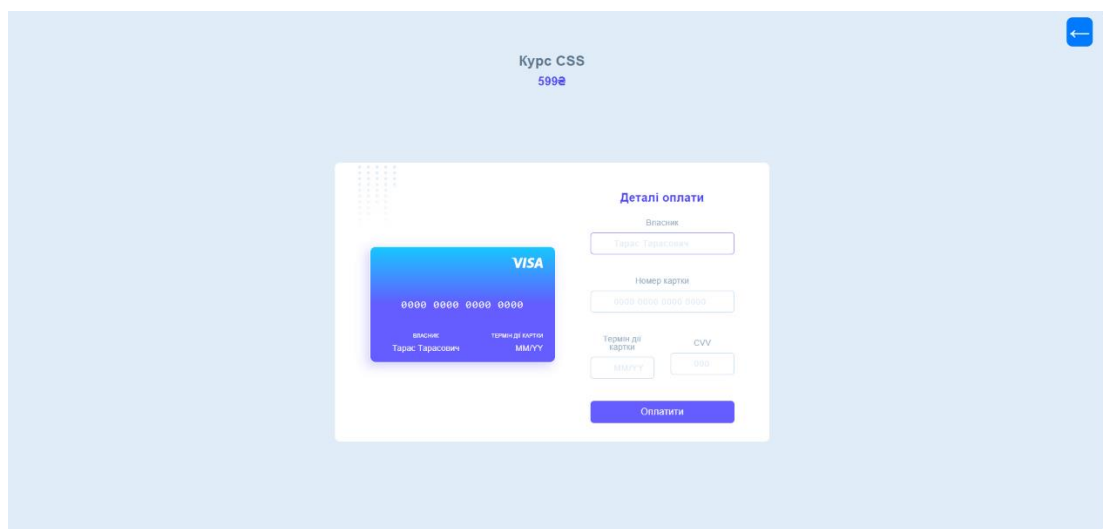


Рисунок 2.2 – Оформлення замовлення на сайті

Серверна частина веб-сайту відповідає за обробку запитів від клієнтів, управління базою даних, а також виконання бізнес-логіки додатку. Для розробки серверної частини було обрано мову програмування PHP, яка забезпечує ефективну роботу з базою даних MySQL. Сервер обробляє запити клієнтів, взаємодіє з базою даних для зберігання та отримання необхідної інформації і відправляє результати назад до клієнта [4].

Коли користувач взаємодіє з веб-сайтом, клієнтська частина відправляє запити до серверної частини за допомогою HTTP-протоколу. Запити можуть бути різних типів, таких як GET, POST, PUT, DELETE, залежно від типу операції, яку необхідно виконати (отримання даних, створення нових записів, оновлення існуючих даних або їх видалення).

На сервері PHP-скрипти обробляють отримані запити, виконують необхідні операції з базою даних MySQL і формують відповіді у вигляді HTML-сторінок або даних у форматі JSON чи XML, які повертаються клієнту. Наприклад, коли користувач оформляє замовлення на курс, відповідний запит

надсилається на сервер, де PHP-скрипт обробляє дані замовлення, зберігає їх у базі даних і повертає підтвердження клієнту.

Безпека є критичним аспектом клієнт-серверної архітектури. Для захисту даних користувачів використовуються різні методи аутентифікації та авторизації, такі як використання сесій і токенів. При вході користувача в систему сервер генерує сесію, яка дозволяє відстежувати стан користувача протягом його взаємодії з веб-сайтом. Це забезпечує безпеку та зручність користувачів, дозволяючи їм залишатися у системі без необхідності повторного введення облікових даних.

Додаткові заходи безпеки включають захист від SQL-ін'єкцій, XSS (міжсайтового скриптингу) та CSRF (підробки міжсайтових запитів). Для цього використовуються підготовлені запити, фільтрація вхідних даних та впровадження спеціальних токенів для підтвердження запитів.

Клієнт-серверна архітектура дозволяє легко масштабувати веб-сайт, забезпечуючи високу продуктивність при збільшенні кількості користувачів та обсягу даних. Серверна частина може бути розширена за рахунок додавання нових серверів або використання хмарних рішень, що дозволяє розподілити навантаження та забезпечити стабільну роботу веб-сайту під високим навантаженням.

2.2 Вибір основних залежностей для розробки веб-сайту для представлення портфоліо

При розробці веб-сайту портфоліо розробника з функціями продажу курсів та надання послуг зі створення веб-сайтів, вибір основних залежностей є критичним етапом. Правильний вибір залежностей забезпечує стабільність,

ефективність та зручність розробки, а також гарантує, що веб-сайт відповідатиме сучасним стандартам і потребам користувачів.

Для фронтенду, ключовими залежностями є ті, що відповідають за створення інтерфейсу користувача і забезпечення інтерактивності веб-сайту. Вибір цих залежностей базується на їх здатності забезпечити легке та швидке створення структурованого і привабливого інтерфейсу. Ці залежності також повинні підтримувати адаптивний дизайн, щоб веб-сайт добре відображався на різних пристроях, включаючи мобільні телефони та планшети.

Фронтенд залежності повинні забезпечувати зручність у використанні і пропонувати готові компоненти, які можна легко інтегрувати у проект, що дозволяє скоротити час розробки. Інструменти для стилізації та анімації повинні надавати можливості для створення привабливого візуального вигляду без необхідності писати великий обсяг коду вручну.

Бекенд залежності є фундаментальними для обробки запитів користувачів, управління даними та реалізації бізнес-логіки додатку. Основні залежності для бекенду повинні забезпечувати стабільну і надійну роботу сервера, ефективну взаємодію з базою даних і підтримку безпеки.

При виборі залежностей для бекенду важливо враховувати їх здатність обробляти великі обсяги даних і підтримувати високу продуктивність. Інструменти для управління базами даних повинні забезпечувати ефективне зберігання і швидкий доступ до даних, а також підтримувати транзакції для забезпечення цілісності даних. Безпека також є ключовим аспектом, тому залежності повинні включати засоби для захисту від поширених веб-атак і управління доступом до даних.

Інструменти, що використовуються для написання коду та управління проектом, також є важливими залежностями. Вибрані інструменти повинні бути зручними у використанні, підтримувати синтаксичне підсвічування, автозаповнення і інші функції, які спрощують процес написання коду. Вони

також повинні підтримувати роботу з кількома мовами програмування, що дозволяє легко інтегрувати різні частини проекту.

Для забезпечення високої якості коду та стабільності веб-сайту необхідні інструменти для тестування та налагодження. Ці інструменти повинні забезпечувати можливості для автоматизованого тестування, виявлення і виправлення помилок, а також моніторингу продуктивності додатку. Вони повинні легко інтегруватися з іншими інструментами розробки і підтримувати сучасні методології розробки

2.3 Файлова структура веб-портфоліо

Файлова структура веб-сайту є важливою складовою для організації коду та ресурсів, що використовуються у проекті. Вона дозволяє чітко розділити логічні компоненти сайту та полегшує навігацію і підтримку коду. Для веб-сайту портфоліо розробника з функціями продажу курсів та надання послуг зі створення веб-сайтів, файлова структура повинна бути інтуїтивно зрозумілою та ефективною. У цьому розділі буде описано основні папки та файли, що входять до структури сайту.

Файлова структура веб-сайту включає такі основні папки та файли: `image`, `img`, `pages`, `script`, `style`, та файл `index.php`. Кожна з цих папок і файлів виконує конкретну функцію в організації сайту.

`Image`. Папка `image` містить усі зображення, що використовуються на веб-сайті. Це можуть бути логотипи, банери, іконки та інші графічні елементи. Зберігання всіх зображень в окремій папці допомагає легко керувати графічним контентом та забезпечує швидкий доступ до необхідних файлів.

`Img`. Папка `img` також призначена для зберігання зображень. Вона може використовуватися для специфічних зображень, пов'язаних з окремими

сторінками або функціями сайту. Наприклад, зображення для галерей, курсів чи профілів користувачів можуть зберігатися у цій папці.

Pages. Папка `pages` містить окремі файли сторінок сайту. Кожна сторінка сайту, така як головна сторінка, сторінка курсів, контактів тощо, має свій окремий файл у цій папці. Це дозволяє структурувати контент і логіку кожної сторінки окремо, що полегшує редагування і підтримку сайту.

Script. Папка `script` містить JavaScript файли, які забезпечують динамічність та інтерактивність веб-сайту. У цій папці зберігаються такі файли:

- `admin.js` – скрипт, що відповідає за функціональність адміністративної частини сайту.

- `payment.js` – скрипт, що обробляє платіжні операції на сайті.

- `script-index.js` – скрипт, що забезпечує функціональність головної сторінки.

- `script-index0overlay.js` – скрипт, що відповідає за накладання елементів на головній сторінці.

- `script-payment.js` – додатковий скрипт для обробки платіжних операцій, який може доповнювати `payment.js`.

style. Папка `style` містить CSS файли, які відповідають за стилізацію веб-сайту. У цій папці зберігаються файли зі стилями, що визначають зовнішній вигляд сторінок, включаючи кольори, шрифти, відступи та розташування елементів. Організація стилів в окремій папці дозволяє легко керувати дизайном сайту і швидко вносити зміни.

index.php. Файл `index.php` є головним файлом сайту, який обробляє основні запити та відповідає за генерацію головної сторінки. Він містить основну логіку сайту, підключення до бази даних та ініціалізацію необхідних ресурсів. `index.php` є точкою входу для більшості запитів, що надходять на сайт.

Вся файлова структура організована таким чином, щоб забезпечити зручність розробки та підтримки веб-сайту. Кожна папка має своє чітке призначення і містить файли, що відповідають за конкретні функції сайту. Це дозволяє легко знаходити і редагувати необхідні файли, а також забезпечує ясність та порядок у проекті (див. рис. 2.3).

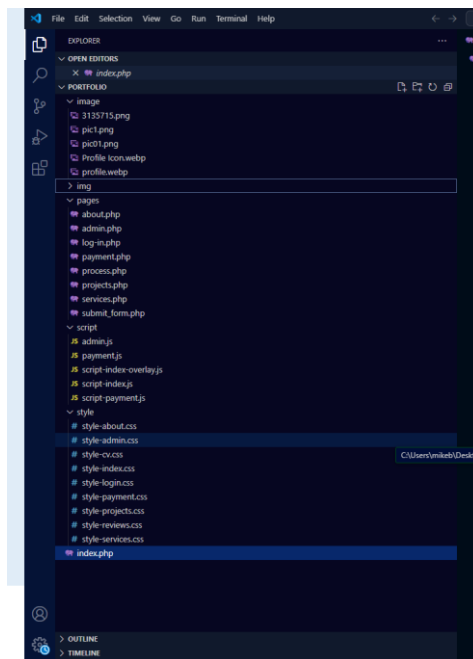


Рисунок 2.3 – Файлова структура проекту

Файли JavaScript у папці script взаємодіють з HTML і PHP кодом, забезпечуючи динамічну поведінку сайту. CSS файли з папки style підключаються до HTML документів, визначаючи їх зовнішній вигляд. Зображення з папок image та img використовуються у різних частинах сайту для забезпечення візуального контенту.

Загалом, файлова структура веб-сайту розроблена таким чином, щоб забезпечити логічну організацію коду та ресурсів, спростити процес розробки і підтримки, а також гарантувати високу продуктивність та зручність використання сайту.

2.4 Структура та особливості використаної БД для розробки веб-портфоліо

Структура бази даних, яка використовується у розробці веб-портфоліо, є ключовою складовою для забезпечення ефективного зберігання та доступу до інформації. База даних є основним механізмом для управління інформацією про користувачів, курси, замовлення та інші важливі аспекти сайту. Для цього проекту була обрана система управління базами даних MySQL, що дозволяє забезпечити надійність та продуктивність. Використання PHPMyAdmin полегшує адміністрування бази даних, надаючи зручний інтерфейс для створення та управління таблицями, а також виконання SQL-запитів [16].

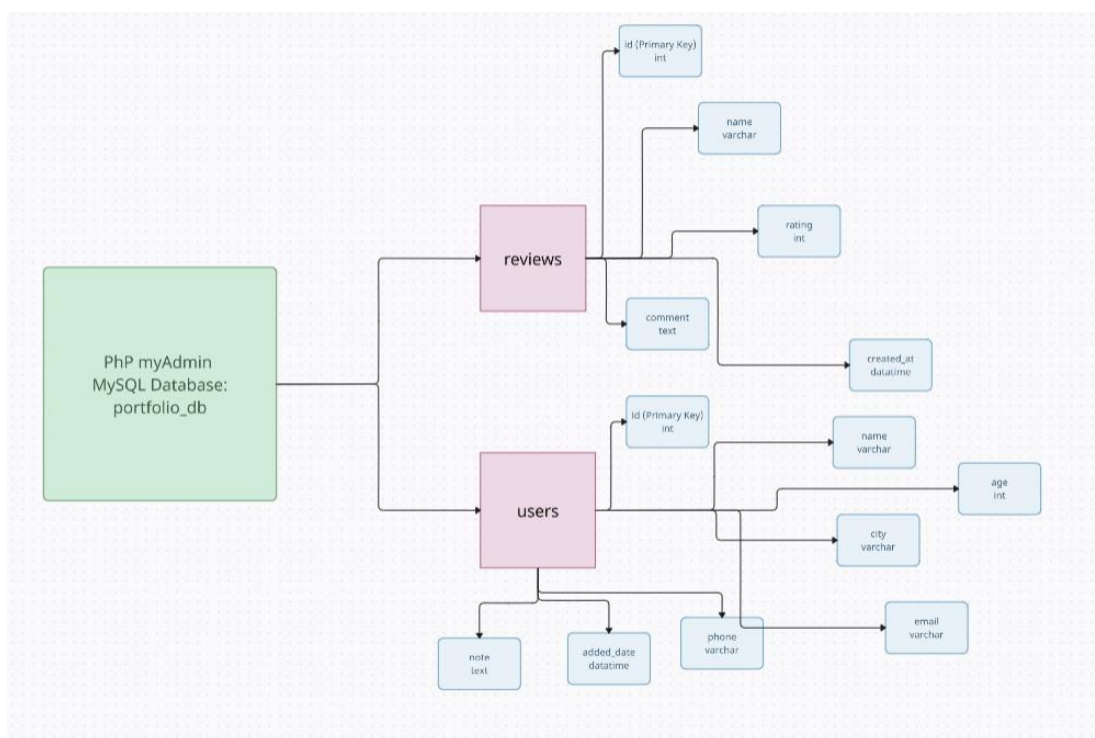


Рисунок 2.4 – Структура БД проекту

У структурі бази даних важливо виділити кілька основних таблиць, кожна з яких відповідає за конкретну категорію даних. Наприклад, таблиця users зберігає інформацію про користувачів сайту, включаючи їхні імена,

паролі, електронні адреси та ролі. Це дозволяє забезпечити персоналізований доступ та управління профілями користувачів, розмежовувати права доступу та забезпечувати безпеку особистих даних. Кожен запис у цій таблиці має унікальний ідентифікатор, що дозволяє однозначно визначити кожного користувача.

Іншою важливою таблицею є reviews, яка містить відгуки користувачів про курси та послуги. Зберігання відгуків у структурованій формі дозволяє легко управляти ними, проводити аналіз якості курсів і реагувати на зворотний зв'язок від користувачів. Кожен відгук асоціюється з конкретним користувачем і курсом, що дозволяє відстежувати якість наданих послуг та забезпечувати прозорість і довіру до платформи.

Таблиці бази даних є основними елементами в реляційних базах даних, які використовуються для зберігання та організації даних у структурованому вигляді. Кожна таблиця складається з рядків і стовпців, де рядки представляють окремі записи, а стовпці визначають типи даних, що зберігаються (див. рис. 2.5).



Рисунок 2.5 – Концептуальна схема бази даних портфоліо

Особливості використання бази даних включають забезпечення цілісності, безпеки та ефективного доступу до даних. Для досягнення цих цілей використовуються різні методи, такі як використання ключових обмежень для забезпечення унікальності та коректності даних, шифрування паролів для захисту особистої інформації, а також регулярне резервне копіювання для збереження даних у разі збоїв. Оптимізація SQL-запитів і використання індексів забезпечують швидкий доступ до необхідної інформації, що є критично важливим для підтримки високої продуктивності сайту.

Таким чином, структура та особливості використаної бази даних забезпечують надійне та ефективне зберігання і управління даними, необхідними для функціонування веб-портфоліо розробника. Це дозволяє реалізувати широкий спектр функціональних можливостей, забезпечуючи зручність для користувачів і надійність роботи платформи.

\

2.5 Адміністрування веб-сайту

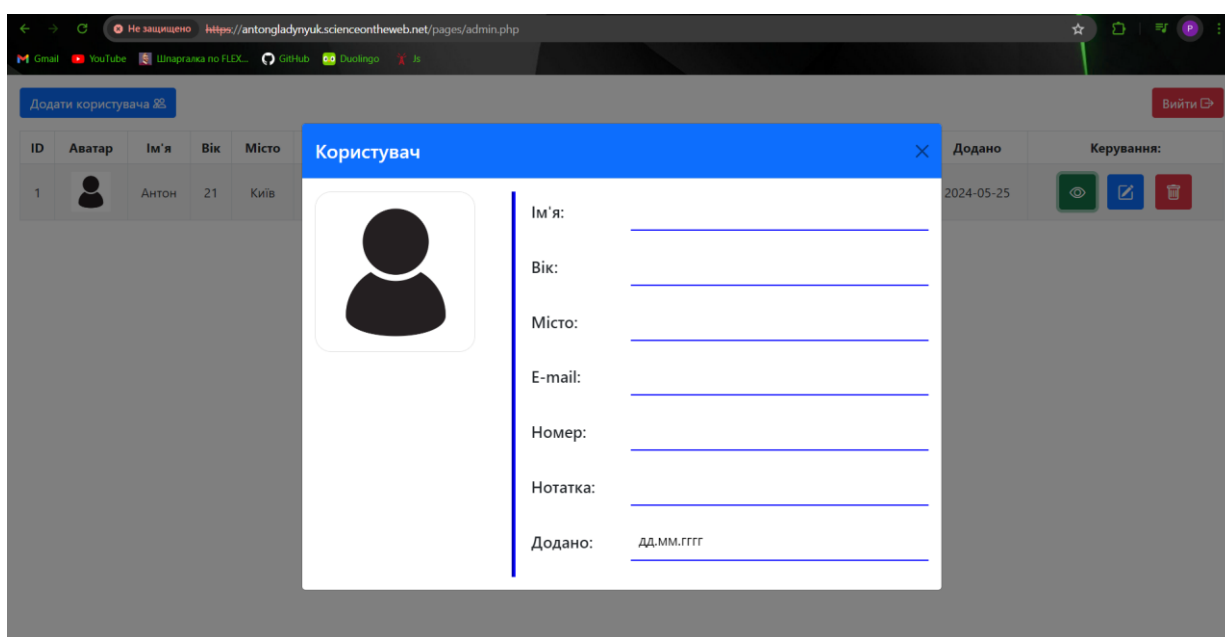
Адміністрування сайту є важливим аспектом його функціонування, оскільки дозволяє підтримувати сайт у належному стані, забезпечувати його актуальність та безпеку. Основним інструментом для адміністрування є адміністративна панель (адмінка), яка надає адміністраторам доступ до різних функцій управління контентом та користувачами. Це зручний інтерфейс, що дозволяє виконувати необхідні завдання без потреби прямого втручання у код сайту.

Адмінка зазвичай реалізована як захищений розділ сайту, доступ до якого мають лише авторизовані користувачі з відповідними правами. Через

адміністративну панель можна додавати, редагувати та видаляти контент, керувати користувачами, стежити за активністю на сайті та виконувати інші адміністративні завдання. Завдяки інтуїтивно зрозумілому інтерфейсу, адміністратори можуть швидко та ефективно вносити зміни, необхідні для підтримки та розвитку веб-ресурсу.

Загалом, адміністрування сайту включає в себе комплекс заходів з управління різними аспектами його роботи. Це може включати оновлення контенту, моніторинг активності користувачів, забезпечення безпеки даних, а також налаштування і оптимізацію різних функцій та компонентів сайту. Ефективне адміністрування дозволяє своєчасно реагувати на потреби користувачів, вносити необхідні зміни та покращення, що в результаті сприяє успішному функціонуванню і розвитку веб-ресурсу.

Конкретний приклад адміністрування портфолію на веб-сайті розробника. В адміністративній панелі сайту є можливість додавати нових користувачів, що дозволяє керувати доступом до різних функцій і розділів сайту. Адміністратор може створити обліковий запис для нового користувача, вказавши його ім'я, електронну пошту, пароль та роль (див. рис. 2.6).



The screenshot displays a web browser window with the URL `https://antongladnyuk.scienceontheweb.net/pages/admin.php`. The page is titled "Додати користувача" (Add user). On the left, there is a table with columns: ID, Аватар, Ім'я, Вік, Місто. The table contains one row with ID 1, a default user icon, the name "Антон", age 21, and city "Київ". A modal form titled "Користувач" is open in the center, containing the following fields:

- Ім'я: (Name)
- Вік: (Age)
- Місто: (Location)
- E-mail:
- Номер: (Number)
- Нотатка: (Note)
- Додано: (Added) with a date format DD.MM.YYYY

On the right side of the modal, there are buttons for "Додано" (Added) with the date "2024-05-25" and "Керування:" (Management) with icons for view, edit, and delete. A "Вийти" (Logout) button is located in the top right corner of the admin panel.

Рисунок 2.6 – Додавання користувача в адмінці

Наприклад, для додавання нового викладача адміністратор може ввести відповідні дані, призначити йому роль "викладач" та надати доступ до функцій управління курсами. Це забезпечує зручне та ефективне управління користувачами і дозволяє розширювати можливості сайту за рахунок залучення нових фахівців.

Адміністративна панель є центральним елементом для управління сайтом, забезпечуючи зручний доступ до основних функцій та інструментів, необхідних для підтримки його працездатності та безпеки. Вона дозволяє адміністраторам легко та ефективно виконувати свої обов'язки, підтримуючи сайт у належному стані та забезпечуючи його стабільну роботу.

2.6 Розгортання веб-сайту портфоліо на хостингових платформах

Розгортання сайту на хостингових платформах є заключним і надзвичайно важливим етапом у процесі створення веб-ресурсу. Хостингова платформа надає інфраструктуру та необхідні сервіси для розміщення сайту в Інтернеті, забезпечуючи його доступність для користувачів з усього світу. Вибір відповідної хостингової платформи значно впливає на продуктивність, безпеку та загальну ефективність сайту [10].

Хостингова платформа надає простір на сервері, де зберігаються файли сайту, бази даних та інші ресурси. Існує кілька типів хостингових послуг, кожен з яких підходить для різних потреб і бюджетів:

— Віртуальний хостинг (shared hosting) — один з найбільш популярних варіантів, де декілька сайтів розміщуються на одному сервері і ділять його ресурси. Це економічний варіант, але може мати обмеження по продуктивності.

— Віртуальний приватний сервер (VPS) – цей варіант передбачає виділення окремого віртуального середовища на сервері, що забезпечує більшу гнучкість та контроль над ресурсами в порівнянні з віртуальним хостингом.

— Виділений сервер (dedicated server) – оренда повного сервера для одного клієнта, що дозволяє максимально контролювати середовище, але є більш коштовним варіантом.

— Хмарний хостинг (cloud hosting) – використовує мережу віртуальних серверів, що дозволяє забезпечити високу масштабованість та надійність, оскільки ресурси можуть бути динамічно розподілені між декількома фізичними машинами.

Для розгортання веб-сайту портфоліо було обрано хостингову платформу scienceontheweb.net. Ця платформа надає всі необхідні сервіси для ефективного розміщення та підтримки веб-ресурсу, включаючи веб-сервери, бази даних, системи безпеки та підтримку доменних імен.

Процес розгортання сайту на scienceontheweb.net включав декілька ключових кроків. Перш за все, було створено обліковий запис на платформі та обрано відповідний тарифний план, що забезпечує необхідні ресурси для роботи сайту. Після цього, було налаштовано доменне ім'я, яке дозволяє користувачам легко знаходити сайт в Інтернеті.

Наступним кроком було завантаження файлів сайту на сервер. Це включає копіювання всіх необхідних файлів, таких як HTML, CSS, JavaScript та PHP скрипти, а також налаштування бази даних. За допомогою інструментів, наданих scienceontheweb.net, таких як FTP-доступ або веб-інтерфейс для завантаження файлів, цей процес був виконаний швидко та ефективно, після чого портфоліо можна побачити в інтернеті (див. рис. 2.7).

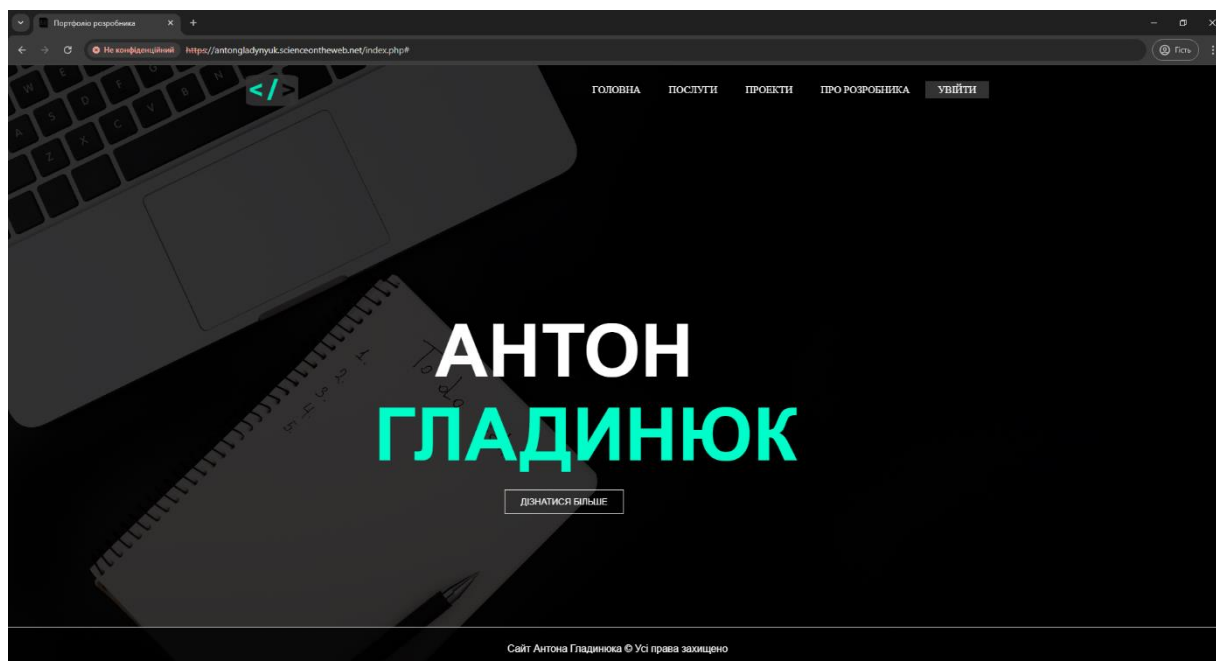


Рисунок 2.7 – Перегляд портфоліо в інтернеті

Одним з важливих етапів було налаштування конфігураційних файлів, таких як `config.php`, де були вказані параметри підключення до бази даних та інші налаштування середовища. Крім того, було налаштовано права доступу до файлів і директорій, що забезпечує безпеку та коректну роботу сайту.

Після завершення всіх налаштувань, сайт був протестований для перевірки його працездатності та усунення можливих помилок. Це включало перевірку всіх основних функцій, таких як реєстрація користувачів, додавання курсів, перегляд портфоліо та здійснення замовлень. Завдяки цьому етапу вдалося переконатися, що сайт працює коректно і готовий до використання користувачами.

2.7 Тестування та використання веб-сайту

2.7.1 Тестування веб-сайту

Тестування сайту є важливим етапом у розробці веб-ресурсу, який допомагає забезпечити його функціональність, надійність та ефективність. Тестування включає різні методи перевірки, які дозволяють виявити і усунути можливі помилки, оптимізувати продуктивність та забезпечити відповідність сайту встановленим вимогам. У випадку нашого веб-сайту портфоліо, тестування охоплювало як функціональні, так і не функціональні аспекти, включаючи перевірку швидкодії та продуктивності.

Функціональне тестування було спрямоване на перевірку всіх основних функцій сайту, щоб переконатися, що вони працюють коректно. Це включало перевірку можливостей реєстрації та авторизації користувачів, додавання та редагування курсів, перегляд портфоліо, здійснення замовлень, а також роботу контактної форми. Тестування проводилось шляхом виконання різних сценаріїв користувача, перевірки введених даних та перевірки відповідей системи на різні дії користувача [5].

Тестування на навантаження дозволяє оцінити, як сайт справляється з великим обсягом користувачів і запитів. Це допомагає виявити можливі проблеми з продуктивністю, які можуть виникнути при реальному використанні сайту. Тестування на навантаження включало імітацію великої кількості одночасних користувачів, що звертаються до сайту, і моніторинг його реакції на підвищене навантаження. Було перевірено, чи справляється сервер із збільшеною кількістю запитів і чи не виникають затримки або збої у роботі.

Оптимізація продуктивності сайту є важливою складовою тестування. Це включає в себе перевірку швидкості завантаження сторінок, ефективність роботи скриптів, а також обробку зображень і медіафайлів. Для перевірки

продуктивності сайту було використано сервіс GTMetrix, який дозволяє оцінити швидкість завантаження сторінок, показники оптимізації та можливі проблеми, що впливають на продуктивність [12].

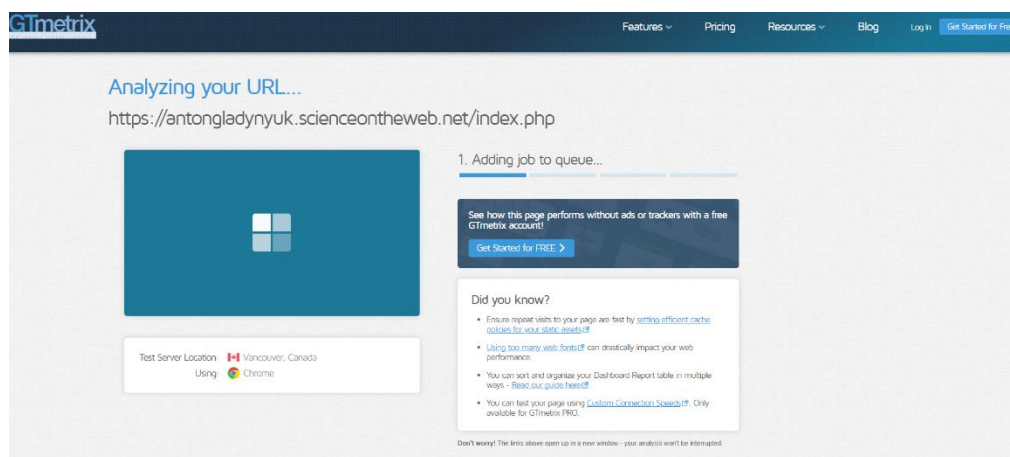


Рисунок 2.8 – Проведення аналізу на Gtmetrix

Після аналізу результатів GTMetrix були виявлені можливі вузькі місця, такі як великі розміри зображень, не оптимізовані скрипти або неефективне використання кешування. Було вжито заходів для оптимізації цих аспектів, що дозволило значно покращити швидкість завантаження сайту (див. рис. 2.9).

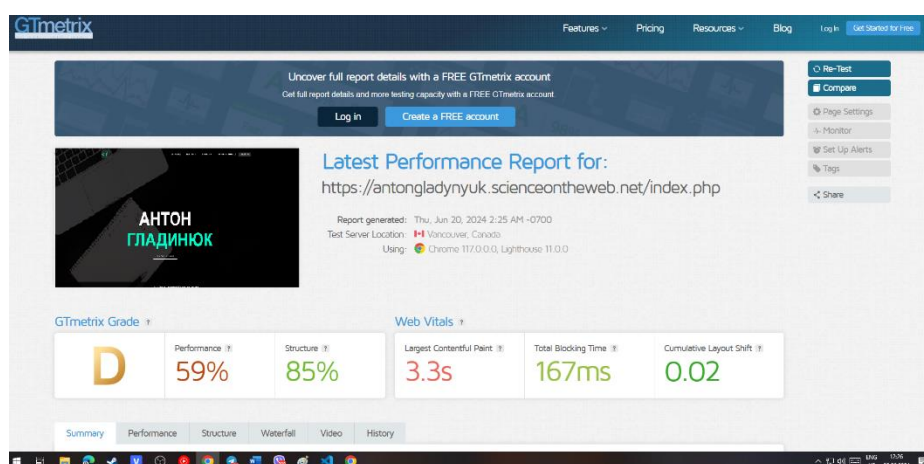


Рисунок 2.9 – Результат аналізу на Gtmetrix

2.7.2 Використання веб-портфоліо

Використання веб-сайту є ключовим аспектом його функціональності, що визначає, як користувачі взаємодіють з ресурсом, отримують необхідну інформацію та виконують дії. Сайт портфоліо розробника містить декілька важливих сторінок, кожна з яких має своє призначення та функціональні можливості, що забезпечують зручність і ефективність для користувачів.

Головна сторінка є стартовою точкою для відвідувачів сайту. Вона представлена як інформаційна панель, яка знайомить користувачів з основними навичками та досвідом розробника. На цій сторінці розміщено блок з коротким описом вмінь розробника, а також кнопку «Дізнатись більше», яка перенаправляє користувача до детальнішої інформації про досвід та кваліфікацію. Такий підхід дозволяє швидко залучити увагу користувачів і надати їм основну інформацію про компетенції розробника [9].

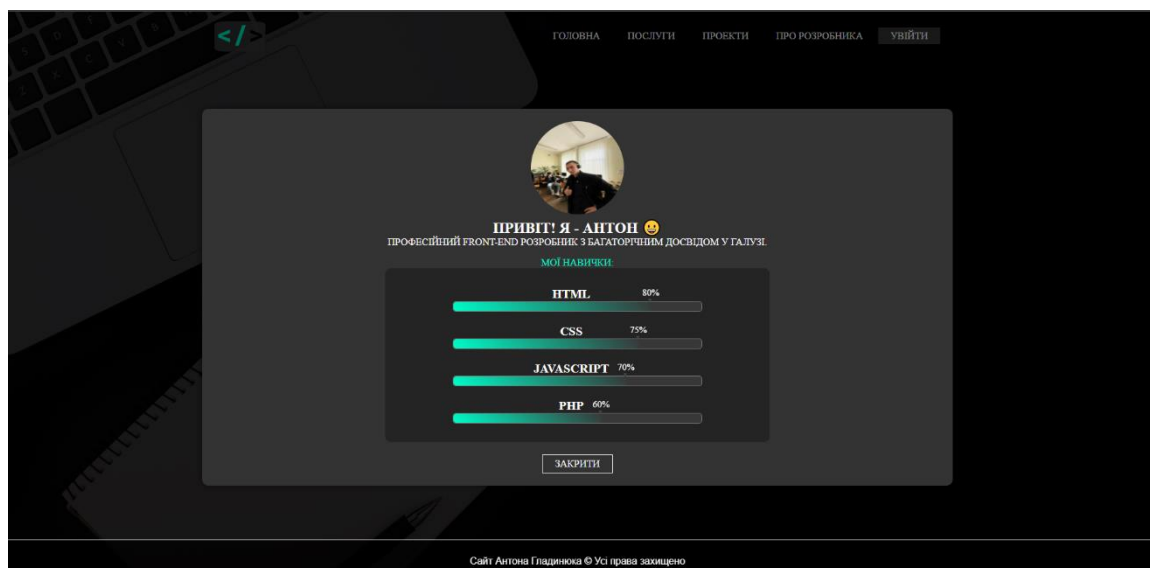


Рисунок 2.10 – Головна сторінка портфоліо

Сторінка з послугами детально описує, які послуги пропонує розробник. Тут користувачі можуть ознайомитись з різноманітними можливостями, які

надаються, такими як розробка веб-сайтів, створення інтерфейсів користувача, оптимізація та підтримка сайтів. Інформація представлена у вигляді переліку послуг з можливістю перегляду деталей кожної послуги, що допомагає користувачам швидко знайти потрібну інформацію та зробити вибір [8].

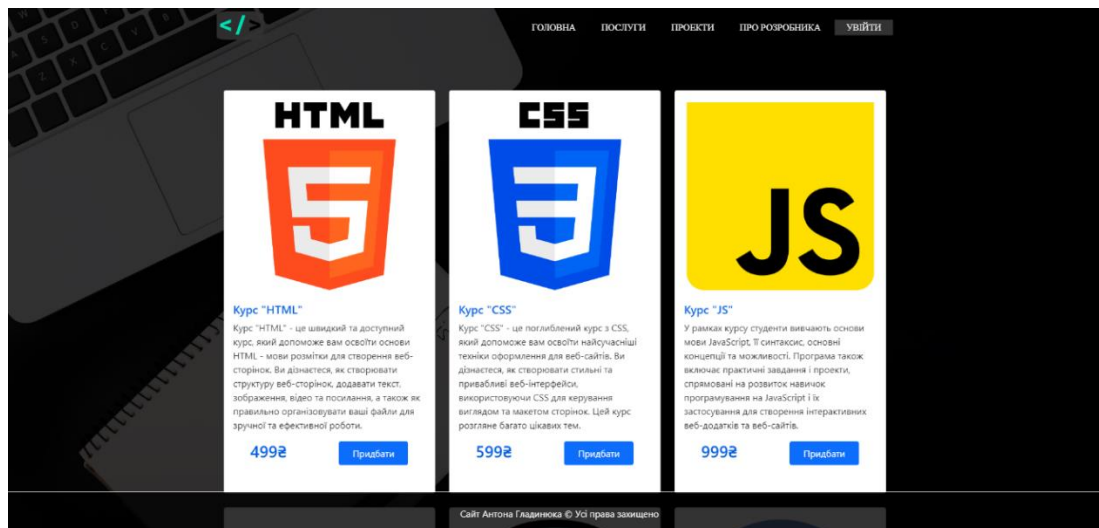


Рисунок 2.11 – Сторінка з послугами

Сторінка портфоліо є однією з основних частин сайту, де представлені минулі проекти розробника. Кожен проект має короткий опис, зображення або демо версію, що дозволяє користувачам переглядати роботи і оцінювати рівень майстерності розробника. Користувачі можуть натискати на окремі проекти, щоб дізнатися більше про кожен з них, включаючи технічні деталі, використані технології та результати проекту [7].

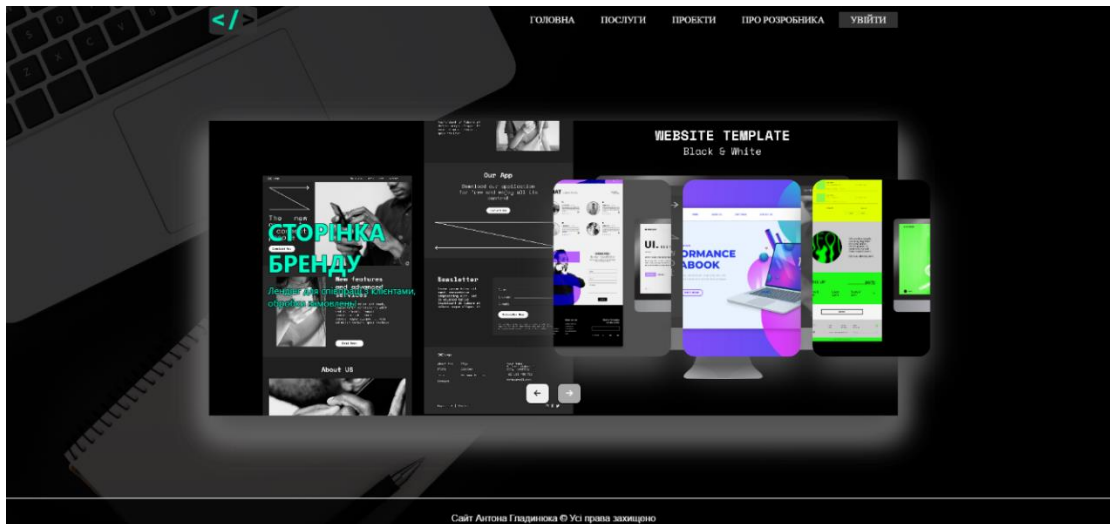


Рисунок 2.12 – Потфоліо на сайті

Сторінка про розробника надає детальну інформацію про особу, що стоїть за проектом. Тут розміщено біографічні дані, професійні досягнення, сертифікати та інші релевантні відомості, що створюють повне уявлення про досвід і кваліфікацію розробника. Ця сторінка також може включати відгуки клієнтів, що додає довіри до професіоналізму розробника [6].

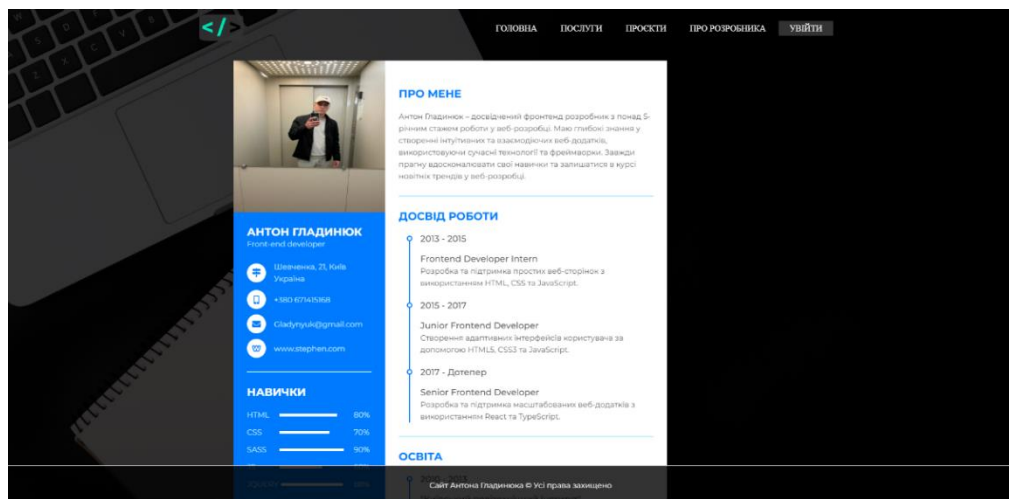


Рисунок 2.13 – Інформація про розробника

Сторінка входу в адмін панель є важливим елементом для користувачів, які мають доступ до адміністративних функцій сайту. Для забезпечення безпеки доступу використовується система аутентифікації, що гарантує, що

тільки уповноважені користувачі можуть виконувати адміністративні дії (див. рис. 2.14).

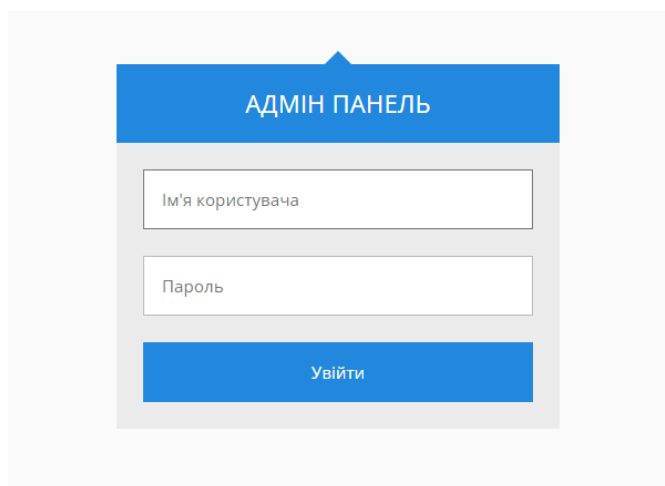
The image shows a login form for an 'Адмін панель' (Admin Panel). It features a blue header with the text 'АДМІН ПАНЕЛЬ'. Below the header, there are two input fields: the first is labeled 'Ім'я користувача' (Username) and the second is labeled 'Пароль' (Password). At the bottom of the form is a blue button labeled 'Увійти' (Login).

Рисунок 2.14 – Вхід в адмін панель

Кожна сторінка сайту має чітко визначене призначення і функціонал, що забезпечує користувачам зручний доступ до необхідної інформації та послуг. Це сприяє покращенню користувацького досвіду і забезпечує ефективне використання сайту для різних цілей, від перегляду портфоліо до отримання консультаційних послуг.

2.8 Висновки до другого розділу

У процесі розробки веб-сайту портфоліо розробника були здійснені важливі кроки, що забезпечили створення ефективного та функціонального ресурсу. Починаючи з аналізу предметної області, ми визначили ключові технології та аспекти, які відповідають вимогам сучасного веб-розробника. Формування вимог до проекту дозволило встановити чіткі цілі та завдання, що стали основою для подальшого планування та реалізації.

При виборі технологій для розробки сайту було обрано PHP та JavaScript без використання фреймворків, що забезпечило достатню гнучкість і контроль над кодом. Під час розробки бази даних було розглянуто кілька варіантів, зупинившись на використанні PHPMyAdmin і MySQL для забезпечення надійного зберігання і обробки даних.

Адміністрування сайту було спроектовано так, щоб забезпечити зручний доступ до управління контентом і користувачами, що дозволяє підтримувати актуальність і безпеку сайту. Розгортання сайту на хостинговій платформі scienceontheweb.net забезпечило його стабільну роботу та доступність для користувачів з усього світу.

Тестування сайту включало перевірку функціональності, продуктивності та безпеки, зокрема тестування на GTMetrix для оптимізації швидкості завантаження сторінок. Це дозволило виявити і усунути можливі проблеми, забезпечивши високий рівень якості і ефективності сайту. Після завершення тестування були виявлені і усунені всі виявлені помилки та недоліки. Оптимізація продуктивності, забезпечення безпеки та виправлення функціональних проблем дозволили забезпечити стабільну, швидку та безпечну роботу сайту. Тестування на GTMetrix підтвердило значне покращення швидкості завантаження сторінок, що позитивно вплинуло на користувацький досвід і загальну ефективність сайту.

Сайт був створений з урахуванням зручності використання, що відображається в структурі сторінок, кожна з яких має конкретне призначення та функціонал, що забезпечує користувачам легкий доступ до потрібної інформації та послуг.

У результаті, створений веб-сайт портфоліо відповідає сучасним вимогам до веб-розробки, забезпечуючи високу якість, безпеку та ефективність, що сприяє успішному представництву розробника в Інтернеті.

3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ

3.1 Психологічні причини нещасних випадків і травматизму

Практика показує, що основні причини травматизму та аварійності найчастіше пов'язані з інженерними недоглядами, недоліками в організації робіт з експлуатації обладнання, неправильною оцінкою стану справ на провадженні, тобто причина закладена в самій людині, так званому «людському факторі». Від психологічної стійкості працівника залежать як його власна безпека, і безпека підприємства у цілому. Тільки за використання комплексного підходу до прогнозування травматизму з урахуванням психологічної стійкості виробничого персоналу можливе суттєве зниження травматизму на підприємстві та підвищення безпеки праці, оскільки психологічна стійкість є тим резервом, який в даний час не до кінця використовується на практиці.

Особлива увага приділяється широкому спектру властивостей особистості, соціальних обставин та виробничих умов праці, що формують психологічні причини свідомого порушення правил безпечної роботи при експлуатації та ремонті сільськогосподарської техніки. Створення максимально безпечних та здорових умов праці є одним із найважливіших завдань. Виконання цього завдання нерозривно пов'язані з удосконаленням методів управління охороною праці з виробництва.

Психологія безпеки розглядає застосування психологічних знань для забезпечення безпеки діяльності людини. Психологія безпеки праці становить важливу ланку в структурі заходів щодо забезпечення безпечної діяльності людини. Проблеми аварійності та травматизму на сучасних виробництвах неможливо вирішувати тільки інженерними методами. В основі аварійності та травматизму лежать організаційно– психологічні причини: низький рівень професійної підготовки з питань безпеки, недостатнє виховання, допуск до

небезпечних видів робіт осіб з підвищеним ризиком травматизації, перебування людей в стані стомлення або інших психічних станах. Таким чином, у психологічній класифікації причин виникнення небезпечних ситуацій і нещасних випадків можна виділити три класи: **Порушення мотиваційної частини дій.** Виявляється в небажанні виконувати певні дії (операції). Порушення може бути відносно постійним (людина недооцінює небезпеку, схильний до ризику, негативно відноситься до трудового та технічного регламенту) і тимчасовим (людина в стані депресії, алкогольного сп'яніння). **Порушення орієнтовної частини дій.** Виявляється в незнанні правила експлуатації сільськогосподарської техніки та технічних систем і норм безпеки праці та способів їх виконання. **Порушення виконавчої частини.** Виявляється в невиконанні правил (інструкцій, розпоряджень, норм) внаслідок невідповідності психічних і фізичних можливостей людини вимогам роботи. Порушення можуть бути постійними (недостатня координація, погана концентрація уваги) і тимчасовими (перевтома, зниження працездатності, погіршення стану здоров'я, стрес, алкогольне сп'яніння). Ці порушення створюють реальну можливість виникнення небезпечних ситуацій і нещасних випадків [14].

Крім того, причини виникнення небезпечних ситуацій і виробничих травм пов'язаних з людським фактором, можна розкласти по різних рівнях, а саме:

- Рівень індивідуума (вроджені чи набуті тимчасово або постійно психічні та фізіологічні характеристики організму);
- Рівень ближній середовища (умови праці, порушення колегіальних відносин, незадовільний інструктаж з безпеки праці, житлові та матеріальні турботи);
- Рівень суспільства (недостатня інформованість про професійні ризики та їх наслідки, вади в стратегії організації безпечної праці в галузі, порушення в умовах праці).

Розглядаючи поведінку працюючого, можна виділити об'єктивні чинники виробничої обстановки, що створюють небезпечні дії і які спричиняють виникнення небезпечних ситуацій:

1. Відсутність з боку адміністрації належного контролю за дотриманням правил безпеки.

2. Конструктивні недоліки засобів захисту працюючих від впливу небезпечних і шкідливих виробничих факторів, які дають можливість проникнення в небезпечну зону.

3. Конструктивні недостачі блокуючих пристроїв, що дозволяють нейтралізувати їх дії і можливість ведення роботи небезпечними методами.

4. Недосконала технологія виробничих процесів, що допускає ведення операції з порушеннями правил безпеки.

5. Плинність кадрів, яка призводить до залучення до робіт у вихідні дні, надурочних робіт; погано поставлена пропагандистська і виховна робота.

6. Порушення правил виконання операцій в результаті відсутності інструменту, механізму, матеріалу, установки, на наявність яких виконавець мав право розраховувати при виконанні завдання.

7. Недостатня надійність. Інструмент, матеріал, верстат або установка раптово стають непридатними для подальшого використання, в той час як раніше вони знаходилися в робочому стані.

8. Недоліки в проектуванні обладнання. Помилки в проектуванні призводять до того, що інструменти, верстати або установки стає важко використовувати при виконанні виробничих завдань.

9. Особисті чинники. Недостатні знання методів виконання завдання, помилки інструктажу, погане навчання способів використання інструменту, неправильна або небезпечна поведінка під час роботи.

10. Неповне виконання елементів завдання. Елемент запланованого завдання не був виконаний вчасно або виконувався з помилками.

11. Небезпечні ситуації, які не залежать від завдання, є потенційною небезпекою (погані умови праці, відсутність або несправність засобів індивідуального захисту). Ризик, пов'язаний з такими ситуаціями, може бути, як правило, виключений за допомогою попереджувальних заходів технічного або організаційного характеру

3.2 Вимоги до профілактичних медичних оглядів для працівників ПК

Порядок проведення медичних оглядів працівників певних категорій (далі - Порядок), розроблений на виконання статті 17 Закону України "Про охорону праці, визначає процедуру проведення попереднього (під час прийняття на роботу) та періодичних (протягом трудової діяльності) медичних оглядів працівників, зайнятих на важких роботах, роботах із шкідливими чи небезпечними умовами праці або таких, де є потреба у професійному доборі, щорічному обов'язковому медичному огляді осіб віком до 21 року [15].

Порядок призначений для: працівників, зайнятих на важких роботах, роботах із шкідливими чи небезпечними умовами праці або таких, де є потреба у професійному доборі, та осіб віком до 21 року підприємств, установ та організацій незалежно від форм власності, виду економічної діяльності та їх філій, інших відокремлених підрозділів; фізичних осіб - суб'єктів підприємницької діяльності, які відповідно до законодавства використовують найману працю (далі - роботодавці); осіб, які забезпечують себе роботою самостійно; закладів державної санітарно-епідеміологічної служби; закладів охорони здоров'я,

військово-лікарських та відповідних комісій міністерств та інших центральних органів виконавчої влади, які здійснюють медичні огляди працівників, спеціалізованих закладів охорони здоров'я, які мають право встановлювати діагноз щодо професійних захворювань, кафедр та курсів професійних захворювань вищих медичних навчальних закладів III-IV рівнів акредитації (далі - закладів охорони здоров'я); робочих органів виконавчої дирекції Фонду соціального страхування від нещасних випадків на виробництві та професійних захворювань України (далі - робочі органи виконавчої дирекції Фонду).

Обов'язкові попередній (під час прийняття на роботу) і періодичні (протягом трудової діяльності) медичні огляди проводяться для працівників, зайнятих на важких роботах, роботах із шкідливими чи небезпечними умовами праці або таких, де є потреба у професійному доборі, та щорічно для осіб віком до 21 року.

Попередній медичний огляд проводиться під час прийняття на роботу з метою:

— визначення стану здоров'я працівника і реєстрації вихідних об'єктивних показників здоров'я та можливості виконання без погіршення стану здоров'я професійних обов'язків в умовах дії конкретних шкідливих та небезпечних факторів виробничого середовища і трудового процесу;

— виявлення професійних захворювань (отруєнь), що виникли раніше при роботі на попередніх виробництвах, та попередження виробничо зумовлених і професійних захворювань (отруєнь).

Періодичні медичні огляди проводяться з метою:

- своєчасного виявлення ранніх ознак гострих і хронічних професійних захворювань (отруєнь), загальних та виробничо зумовлених захворювань у працівників;

- забезпечення динамічного спостереження за станом здоров'я працівників в умовах дії шкідливих та небезпечних виробничих факторів і трудового процесу;

- вирішення питання щодо можливості працівника продовжувати роботу в умовах дії конкретних шкідливих та небезпечних виробничих факторів і трудового процесу;

- розробки індивідуальних та групових закладів охорони здоров'я та реабілітаційних заходів працівникам, що віднесені за результатами медичного огляду до групи ризику;

- проведення відповідних оздоровчих заходів.

Хороший роботодавець повинен піклуватися про здоров'я своїх працівників. Адже здорові працівники — це запорука злагодженої роботи всієї установи. «Стимулюють» роботодавця не забувати про здоров'я працівників законодавство про охорону праці та КЗпП.

Для кого обов'язкові трудові медогляди:

- Для працівників зайнятих на важких роботах.
- Зі шкідливими або небезпечними умовами праці.
- Де є необхідність у професійному доборі.
- Для водіїв транспортних засобів.
- Молодше 18 років (при прийнятті на роботу).
- Молодше за 21 рік (щорічно).

Чи можна обійтися без медогляду? Ні, і ще раз ні! Якщо звернутися до КЗпП, то можна побачити, що:

- роботодавець зобов'язаний за свої кошти організувати проведення попереднього (при прийнятті на роботу) і періодичних (протягом трудової діяльності) медоглядів певних категорій працівників (див. вище) (ст. 169 КЗпП);

— працівник зобов'язаний проходити в установленому порядку попередні та періодичні медогляди (ст. 159 КЗпП).

— Про обов'язкові профілактичні медогляди працівників деяких професій, виробництв та організацій ідеться в ст. 21 Закону № 1645.

Таким чином, трудові і профілактичні медогляди — обов'язкова законодавча умова, дотримання якої надає право:

— роботодавцеві — допустити працівників до виконання їх трудових обов'язків;

— працівникам — виконувати свою роботу.

Тож відповідь однозначна: без медогляду ніяк не обійтися. У випадку якщо працівник відмовляється проходити медогляд, у роботодавця з'являються підстави для відсторонення його від роботи без збереження зарплати і притягнення до дисциплінарної відповідальності.

3.3 Висновок до третього розділу

У розділі з охорони праці для проекту веб-сайту портфоліо було розглянуто два основні питання: обов'язкові медичні огляди працівників певних категорій та психологічні причини нещасних випадків і трав на виробництві. Ці аспекти є важливими для забезпечення безпеки та здоров'я працівників, які займаються розробкою та підтримкою веб-сайту.

Медичні огляди працівників, особливо тих, хто виконує роботи, пов'язані з підвищеними ризиками для здоров'я, є необхідним заходом для своєчасного виявлення та попередження професійних захворювань. У проекті передбачено регулярне проведення обов'язкових медичних оглядів для працівників, що працюють з комп'ютерним обладнанням та програмним забезпеченням протягом тривалого часу. Це допомагає виявляти потенційні

проблеми зі здоров'ям, такі як захворювання опорно-рухового апарату, зорові порушення та інші професійні хвороби, а також своєчасно вживати необхідних заходів для їх запобігання.

У розділі, присвяченому психологічним причинам нещасних випадків на виробництві, ми виявили, що стан психологічного здоров'я працівників має вагоме значення для забезпечення їх безпеки. Фактори, такі як стрес, втома, недостатня мотивація та недооцінка ризиків, можуть значно підвищувати ймовірність виникнення аварійних ситуацій. Психологічні аспекти, включаючи емоційний стан, рівень стресу та комунікацію в колективі, прямо впливають на здатність працівників адекватно реагувати в екстрених ситуаціях.

Проект веб-сайту портфоліо включає системний підхід до охорони праці, який охоплює не лише вищезазначені питання, але й загальну підтримку здорового способу життя серед працівників. Важливим аспектом є створення комфортного робочого середовища, яке включає ергономічне обладнання, оптимальне освітлення та вентиляцію робочих місць, а також можливість для працівників відпочивати і займатися фізичною активністю.

Для працівників, що займаються творчою та інтелектуальною працею, важливим є забезпечення психологічного комфорту. Це досягається через створення позитивної робочої атмосфери, підтримку відкритого спілкування та співпраці між членами команди, а також проведення тренінгів та заходів, спрямованих на зниження стресу та підвищення мотивації.

Компанія прагне до постійного вдосконалення умов праці шляхом впровадження новітніх технологій та методик, що дозволяють зменшити негативний вплив роботи на здоров'я працівників. Використання сучасних засобів захисту, регулярний моніторинг стану здоров'я працівників та аналіз умов праці сприяють підтриманню високого рівня безпеки та здоров'я на робочих місцях.

У підсумку, розділ з охорони праці в проекті веб-сайту портфоліо демонструє комплексний підхід до забезпечення безпечних та комфортних умов праці. Всі заходи, що впроваджуються, спрямовані на підтримання фізичного та психологічного здоров'я працівників, підвищення їхньої продуктивності та задоволеності роботою. Цей підхід дозволяє створити здорове та ефективне робоче середовище, що є ключовим для успішної реалізації проекту.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи освітнього рівня “бакалавр”, розробки веб-портфоліо розробника такими технологіями як: Php, JavaScript, HTML та CSS, було виконано всі раніше поставлені завдання при дослідженні задачі.

Результати аналізу теми були оформлені у вигляді трьох частин кваліфікаційної роботи, з яких дві є технічними і спрямовані на створення веб-сайту.

В першому розділі було проаналізовано предметну область, сформовано вимоги до розробки веб-сайту, проведено пошук актантів та можливі варіанти використання веб-сайту, детальніше описано ключові. Вибрано стек для реалізації.

В другому розділі описано архітектуру веб-портфоліо детальніше про вибір технологій відповідно до стеку, файлову структуру веб-сайту, структурні особливості вибраної бази даних, розгортання на хостингових платформах, про тестування та використання розробленої системи.

В третьому розділі, було розглянуто психологічні причини нещасних випадків та травматизму та вимоги до медичних оглядів працівників за ПК

В процесі роботи над кваліфікаційною роботою:

- здобуто нові теоретичні знання з теми дослідження, усунуто наявні прогалини у знаннях;
- вивчено сучасні інструменти для розробки веб-сайтів і обрано найбільш підходящий;
- проведено аналіз функціональних можливостей веб-сайту порівняно з конкурентами;

— на основі отриманої інформації веб-сайт було розроблено та випробувано.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Введення до HTML ТА CSS [Електронний ресурс] – 19.06. 2024 – Режим доступу до ресурсу: <https://promoter.net.ua/articles/osnovi-html-ta-css-stvorennia-persix-storinok.html>
2. JavaScrip Docs [Електронний ресурс] – 19.06. 2024 – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
3. Netepad++ docs [Електронний ресурс] – 19.06. 2024 – Режим доступу до ресурсу: <http://www.lexikoukr.ho.ua/Np%2B%2Bmanual/documentation/notepad-online-document.html>
4. PHP docs [Електронний ресурс] – 19.06. 2024 – Режим доступу до ресурсу: <https://www.php.net/docs.php>
5. Визначення функціонального тестування [Електронний ресурс] – 19.06. 2024– Режим доступу до ресурсу: <https://qalight.ua/baza-znaniy/funktsionalne-testuvannya/>.
6. Сторінка з портфоліо на сайті [Електронний ресурс] – 19.06. 2024 – Режим доступу до ресурсу: <https://antongladynyuk.scienceontheweb.net/pages/about.php>
7. Сторінка з проектами на сайті [Електронний ресурс] – 19.06. 2024 – Режим доступу до ресурсу: <https://antongladynyuk.scienceontheweb.net/pages/projects.php>
8. Сторінка з послугами на сайті [Електронний ресурс] – 19.06. 2024 – Режим доступу до ресурсу: <https://antongladynyuk.scienceontheweb.net/pages/services.php>
9. Головна сторінка портфоліо [Електронний ресурс] – 19.06. 2024– Режим доступу до ресурсу: <https://antongladynyuk.scienceontheweb.net/>
10. Хостинг і домен [Електронний ресурс] – 19.06. 2024– Режим доступу до ресурсу: <https://scienceontheweb.net/>

11. Scienceontheweb документація [Електронний ресурс] – 19.06. 2024 – Режим доступу до ресурсу: <https://github.com/ndiwi2013/aipad.scienceontheweb.net>
12. Документація по експлуатації gtmetrix [Електронний ресурс] – 19.06. 2024 – Режим доступу до ресурсу: <https://gtmetrix.com/api/docs/2.0/>
13. Тестування сайту [Електронний ресурс] – 19.06. 2024 – Режим доступу до ресурсу: <https://gtmetrix.com/>
14. Підручник з безпеки життєдіяльності та охорони праці [Електронний ресурс]. – Режим доступу до ресурсу: https://eprints.kname.edu.ua/57820/1/2019_3%D0%9F_%D0%A1%D0%B1%D0%BE%D1%80%D0%BA%D0%B0%20%D0%91%D0%96%D0%94%D0%9E%D0%9F%20%D0%91%D0%96%D0%94.pdf
15. Навчальний посібник з охорони праці [Електронний ресурс]. Режим доступу до ресурсу: <https://kpdi.edu.ua/biblioteka/%D0%91/%D0%91%D0%96%D0%94%20%D0%97%D0%B0%D1%86%D0%B0%D1%80%D0%BD%D0%B8%D0%B9%20%D0%92.%D0%92.pdf>
16. PhpMyAdmin documentation [Електронний ресурс] – 19.06. 2024 – Режим доступу до ресурсу: <https://www.phpmyadmin.net/docs/>

ДОДАТКИ

Лістинг 1 – Код файлу «index.php»

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    <title>Портфоліо розробника</title>

        <link      rel="icon"      href="/img/title.png"
type="image/png">

        <link      rel="stylesheet"  href="/style/style-
index.css">

    </head>

<body>

<header>

    <div class="wrapper">

        <div class="logo">

        </div>

        <ul class="nav-area">

            <li><a href="/index.php">Головна</a></li>

            <li><a href="/pages/services.php">Послуги</a></li>

            <li><a href="/pages/projects.php">Проекти</a></li>
```

```

<li><a href="/pages/about.php">Про розробника</a></li>
<li><a href="/pages/log-in.php">Увійти</a></li>
</ul>
</div>

<div class="welcome-text">

    <h1>

        Антон <span> Гладинюк</span></h1>

        <a href="#" onclick="openPopup()">Дізнатися
більше</a>

    </div>

<div id="overlay">

    <div id="popup">

        <h3>Привіт! Я - Антон &#128512</h3>

        <p>Професійний Front-end розробник з
багаторічним досвідом у галузі.</p>

        <span class="text">Мої навички:</span>

        <div class="skill_bar">

            <div class="skill_line">

                <span>HTML</span>

                <div class="progress">

                    <div class="progress_line
html"></div>

```

```

        </div>
    </div>
    <div class="skill_line">
        <span>CSS</span>
        <div class="progress">
            <div class="progress_line css"></div>
        </div>
    </div>
    <div class="skill_line">
        <span>JavaScript</span>
        <div class="progress">
            <div class="progress_line js"></div>
        </div>
    </div>
    <div class="skill_line">
        <span>PHP</span>
        <div class="progress">
            <div class="progress_line php"></div>
        </div>
    </div>
</div>
<button onclick="closePopup()">Закрити</button>
</div>
</div>

```

```
<footer>

<div class="line"></div>

<div class="footer-content">

    <p>Сайт Антона Гладинюка © Усі права захищено</p>

</div>

</footer>

</header>


<script src="/script/script-index.js"></script>
<script src="/script/script-index-overlay.js"></script>

</body>

</html>
```

Лістинг 2 – Код файлу «admin.js»

```
document.addEventListener("DOMContentLoaded",
function() {
    let form = document.getElementById("myForm"),
        imgInput = document.querySelector(".img"),
        file = document.getElementById("imgInput"),
        userName = document.getElementById("userName"),
        age = document.getElementById("age"),
        city = document.getElementById("city"),
        email = document.getElementById("email"),
        phone = document.getElementById("phone"),
        post = document.getElementById("post"),
        sDate = document.getElementById("sDate"),
        submitBtn = document.querySelector(".submit"),
        userInfo = document.getElementById("data"),
        modal = document.getElementById("userForm"),
        modalTitle = document.querySelector("#userForm
.modal-title"),
        newUserBtn = document.querySelector(".newUser");

    document.getElementById("logoutBtn").addEventListener(
"click", function() {
        window.location.href = "log-in.php";
```



```

    });

    let getData = localStorage.getItem('userProfile') ?
JSON.parse(localStorage.getItem('userProfile')) : []

    let isEdit = false, editId;

    showInfo();

    newUserBtn.addEventListener('click', () => {
        submitBtn.innerText = 'Підтвердити';
        modalTitle.innerText = "Заповніть форму";
        isEdit = false;
        imgInput.src = "/image/Profile Icon.webp";
        form.reset();
    });

    file.onChange = function(){
        if(file.files[0].size < 1000000){    // 1MB =
1000000

        var fileReader = new FileReader();

        fileReader.onload = function(e){
            imgUrl = e.target.result;
            imgInput.src = imgUrl;
        }
    }

```

```

        fileReader.readAsDataURL(file.files[0]);
    }
    else{
        alert("Цей файл занадто великий!");
    }
}

function showInfo(){
    document.querySelectorAll('.employeeDetails').
    forEach(info => info.remove());
    getData.forEach((element, index) => {
        let createElement = `<tr
class="employeeDetails">
    <td>${index+1}</td>
    <td></td>
    <td>${element.employeeName}</td>
    <td>${element.employeeAge}</td>
    <td>${element.employeeCity}</td>
    <td>${element.employeeEmail}</td>
    <td>${element.employeePhone}</td>
    <td>${element.employeePost}</td>
    <td>${element.startDate}</td>
    <td>

```

```

<button class="btn btn-success"
onclick="readInfo('${element.picture}',
'${element.employeeName}',          '${element.employeeAge}',
'${element.employeeCity}',          '${element.employeeEmail}',
'${element.employeePhone}',          '${element.employeePost}',
'${element.startDate}')" data-bs-toggle="modal" data-bs-
target="#readData"><i class="bi bi-eye"></i></button>

```

```

<button class="btn btn-primary"
onclick="editInfo(${index},          '${element.picture}',
'${element.employeeName}',          '${element.employeeAge}',
'${element.employeeCity}',          '${element.employeeEmail}',
'${element.employeePhone}',          '${element.employeePost}',
'${element.startDate}')" data-bs-toggle="modal" data-bs-
target="#userForm"><i class="bi bi-pencil-
square"></i></button>

```

```

<button class="btn btn-danger"
onclick="deleteInfo(${index})"><i class="bi bi-
trash"></i></button>

```

```

</td>

```

```

</tr>`;

```

```

userInfo.innerHTML += createElement;

```

```

});

```

```

}

```

```

window.readInfo = function(pic, name, age, city,
email, phone, post, sDate) {

```

```

    document.querySelector('.showImg').src = pic;

```

```

        document.querySelector('#showName').value =
name;

        document.querySelector("#showAge").value = age;

        document.querySelector("#showCity").value =
city;

        document.querySelector("#showEmail").value =
email;

        document.querySelector("#showPhone").value =
phone;

        document.querySelector("#showPost").value =
post;

        document.querySelector("#showsDate").value =
sDate;

    }

```

```

        window.editInfo = function(index, pic, name, Age,
City, Email, Phone, Post, Sdate) {

            isEdit = true;

            editId = index;

            imgInput.src = pic;

            userName.value = name;

            age.value = Age;

            city.value = City;

            email.value = Email;

            phone.value = Phone;

```

```

    post.value = Post;

    sDate.value = Sdate;

    submitBtn.innerText = "Оновити";
    modalTitle.innerText = "Оновити форму";
}

window.deleteInfo = function(index) {
    if(confirm("Ви впевнені, що хочете видалити?")) {
        getData.splice(index, 1);
        localStorage.setItem("userProfile",
JSON.stringify(getData));
        showInfo();
    }
}

form.addEventListener('submit', (e) => {
    e.preventDefault();

    const information = {
        picture: imgInput.src == undefined ?
"/image/Profile Icon.webp" : imgInput.src,
        employeeName: userName.value,
        employeeAge: age.value,
        employeeCity: city.value,

```

```

        employeeEmail: email.value,
        employeePhone: phone.value,
        employeePost: post.value,
        startDate: sDate.value
    }

    if(!isEdit){
        getData.push(information);
    }
    else{
        isEdit = false;
        getData[editId] = information;
    }

        localStorage.setItem('userProfile',
JSON.stringify(getData));

    submitBtn.innerText = "Підтвердити";
    modalTitle.innerHTML = "Заповніть форму";

    showInfo();

    form.reset();

    imgInput.src = "/image/Profile Icon.webp";
}))

```

Лістинг 3 – Код файлу «script-index.js»

```
document.addEventListener('DOMContentLoaded',
function() {
    const buyButtons = document.querySelectorAll('.btn-
primary');
    buyButtons.forEach(button => {
        button.addEventListener('click', function() {
            const courseName = this.getAttribute('data-
course');
            const coursePrice = this.getAttribute('data-
price');
            window.location.href =
`payment.php?course=${courseName}&price=${coursePrice}`;
        });
    });
});

document.getElementsByTagName("h1")[0].style.fontSize =
"6vw";
```

Лістинг 4 – Код файлу «script-payment.js»

```
const form = document.querySelector('.form');  
const name = document.getElementById('name');  
const number = document.getElementById('number');  
const date = document.getElementById('date');  
const cvv = document.getElementById('cvv');  
  
const visa = document.querySelector('.card');  
  
function showError(element, error) {  
    if(error === true) {  
        element.style.opacity = '1';  
    } else {  
        element.style.opacity = '0';  
    }  
};  
  
name.addEventListener('input', function() {  
    let alert1 = document.getElementById('alert-1');  
    let error = this.value === '';  
    showError(alert1, error);  
    document.getElementById('card-name').textContent =  
this.value;  
});
```



```

number.addEventListener('input', function(e) {
    this.value = numberAutoFormat();

    let error = this.value.length !== 19;
    let alert2 = document.getElementById('alert-2');
    showError(alert2, error);

    document.querySelector('.card__number').textContent
= this.value;
});

function numberAutoFormat() {
    let valueNumber = number.value;

    let v = valueNumber.replace(/\s+/g,
'').replace(/^[^0-9]/gi, '');

    let matches = v.match(/\d{4,16}/g);
    let match = matches && matches[0] || '';
    let parts = [];

    for (i = 0; i < match.length; i += 4) {

```

```

        parts.push(match.substring(i, i + 4));
    }

    if (parts.length) {

        return parts.join(' ');
    } else {
        return valueNumber;
    }
};

date.addEventListener('input', function(e) {

    this.value = dateAutoFormat();

    let alert3 = document.getElementById('alert-3');
    showError(alert3, isNotDate(this));

    let dateNumber = date.value.match(/\d{2,4}/g);

    document.getElementById('month').textContent =
dateNumber[0];

    document.getElementById('year').textContent =
dateNumber[1];

});

```

```

function isNotDate(element) {
    let actualDate = new Date();
    let month = actualDate.getMonth() + 1;
    let year = Number(actualDate.getFullYear().toString().substr(-2));
    let dateNumber = element.value.match(/\d{2,4}/g);
    let monthNumber = Number(dateNumber[0]);
    let yearNumber = Number(dateNumber[1]);

    if(element.value === '' || monthNumber < 1 ||
monthNumber > 12 || yearNumber < year || (monthNumber <=
month && yearNumber === year)) {
        return true;
    } else {
        return false;
    }
}

```

```

function dateAutoFormat() {
    let dateValue = date.value;

    let v = dateValue.replace(/\s+/g, '').replace(/[^0-
9]/gi, '');

```

```

    let matches = v.match(/\d{2,4}/g);
    let match = matches && matches[0] || '';
    let parts = [];

    for (i = 0; i < match.length; i += 2) {

        parts.push(match.substring(i, i + 2));
    }

    if (parts.length) {

        return parts.join('/');
    } else {

        return dateValue;
    }
};

cvv.addEventListener('input', function(e) {
    let alert4 = document.getElementById('alert-4');
    let error = this.value.length < 3;
    showError(alert4, error)
});

```

```

function isNumeric(event) {
    if ((event.keyCode < 48 || event.keyCode > 57) &&
(event.keyCode > 31)) {
        return false;
    }
};

form.addEventListener('submit', function (e) {

    if(name.value === '' || number.value.length !== 19
|| date.value.length !== 5 || isNotDate(date) === true ||
cvv.value.length < 3) {

        e.preventDefault();

    };

    let input = document.querySelectorAll('input');
    for( i = 0; i < input.length; i++) {
        if(input[i].value === '') {
            input[i].nextElementSibling.style.opacity =
'1';

        }
    }

});

```

Додаток Д
Диск з роботою