

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

КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

РОЗРОБКА САЙТУ БУДІВЕЛЬНОЇ ФІРМИ ТОВ
«ІМПЕРІУМДЕВЕЛОПМЕНТ» З ВИКОРИСТАННЯМ PHP-
ФРЕЙМВОРКУ LARAVEL

DEVELOPMENT OF THE SITE OF THE CONSTRUCTION
FIRM «IMPERIUMDEVELOPMENT» LLC USING THE
LARAVEL PHP FRAMEWORK

спеціальність 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

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

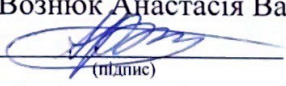
Виконав: здобувач вищої освіти
Групи ІПЗс-21

Дячук Василь Васильович


(підпис)

Керівник:
асистент

Вознюк Анастасія Вадимівна


(підпис)

Кваліфікаційну роботу
допущено до захисту

«2» 06 2024 р.

к.т.н., доцент

Гарант освітньої програми:

Ліщина Наталія Миколаївна


(підпис)

Луцьк – 2024 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти: бакалавр

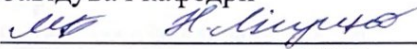
Галузь знань: 12 Інформаційні технології

Спеціальність: 121 Інженерія програмного забезпечення

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри



« 2 » 02 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Дячуку Василю Васильовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Розробка сайту будівельної фірми ТОВ

«ІМПЕРІУМДЕВЕЛОПМЕНТ» з використанням PHP-фреймворку Laravel

Керівник роботи: асистент, Вознюк Анастасія Вадимівна

затверджені наказом закладу вищої освіти від «30» грудня 2023 р. № 477/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи «5» червня 2024 р.

3. Вихідні дані до роботи: технічне та програмне забезпечення ЕОМ, вимоги до розробки об'єкту проектування.

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

аналіз розробки сайтів для будівельних компаній і постановка завдання кваліфікаційної

Роботи; специфікація вимог до розроблювальної системи; розробка сайту будівельної

Компанії «ІМПЕРІУМДЕВЕЛОПМЕНТ»

5. Перелік графічного матеріалу: 22 рисунків

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та по-сада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Вознюк А. В.		
Специфікації вимог до розробле-ної системи	Вознюк А. В.		
Розробка об'єкта проєктування	Вознюк А. В.		
Нормоконтроль	Повстяна Ю. С.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту		7,4 %	
Академічна доброчесність	Яцук А. А.		

7. Дата видачі завдання «2» лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ З/п	Назва етапів кваліфікаційної роботи бакалавра	Строк вико- нання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфіка-ційної роботи бакалавра	07.02.2024 р.	Зачислено
2	Аналіз проблеми розробки та впровадження об'єкту проєктування	27.02.2024 р.	Зачислено
3	Обґрунтування вибору шляхів, технологій (алго-ритмів) і засобів вирішення поставленого за-вдання	12.03.2024 р.	Зачислено
4	Розробка функціонально-структурної схеми ро-боти об'єкта проєктування та проєктування бази даних	17.04.2024 р.	Зачислено
5	Практична реалізація об'єкта проєктування та розробка бази даних	18.05.2024 р.	Зачислено
6	Тестування та налагодження об'єкта проєкту-вання	01.06.2024 р.	Зачислено
7	Здача чистового варіанту кваліфікаційної ро-боти бакалавра на кафедру	05.06.2024 р.	Зачислено

Здобувач вищої освіти

Керівник кваліфікаційної роботи

(підпис)

(підпис)

Дячук В. В.
(прізвище, ініціали)

Вознюк А. В.
(прізвище, ініціали)

Міністерство освіти і науки України
Луцький національний технічний університет

**Рецензія
на кваліфікаційну роботу**

Здобувач вищої освіти: Дячук Василь Васильович

Тема: Розробка сайту будівельної фірми ТОВ «ІМПЕРІУМДЕВЕЛОПМЕНТ» з використанням PHP-фреймворку Laravel

Коротка характеристика кваліфікаційної роботи: Робота присвячена розробці сайту будівельної фірми ТОВ «ІМПЕРІУМДЕВЕЛОПМЕНТ». Створений сайт надає зручний і ефективний інструмент для користувачів та адміністраторів, відповідаючи сучасним стандартам функціональності та дизайну. Розробка сайту була здійснена за допомогою фреймворку Laravel, з використанням PHP та JavaScript.

Самостійні розробки і пропозиції автора: У роботі детально викладено процес створення сайту будівельної фірми ТОВ «ІМПЕРІУМДЕВЕЛОПМЕНТ» та доведено ефективність обраних засобів та методів розробки.

Практичне значення роботи: Сайт будівельної фірми ТОВ «ІМПЕРІУМДЕВЕЛОПМЕНТ» був успішно реалізований і є важливим інструментом для сприяння комунікації зі спільнотою. Сайт протестовано в умовах, наближених до реальних. Кваліфікаційна робота має значне практичне значення.

Недоліки: В роботі відсутні рекомендації щодо подальшого розвитку та масштабування сайту.

Загальний висновок: Робота виконана на високому науковому рівні, має логічну та послідовну структуру, відповідає встановленим вимогам, може бути представлена для захисту на державній екзаменаційній комісії та заслуговує на позитивну оцінку.

Рецензент: к.т.н., доцент, Редько Ольга Іванівна

Рецензент кваліфікаційної роботи _____


(підпис)

«07» серпня 2024 р.

Міністерство освіти і науки України
Луцький національний технічний університет

Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення

ВІДГУК
КЕРІВНИКА НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА
Здобувач вищої освіти Дячук Василь Васильович
група ППЗс-21

Тема кваліфікаційної роботи: Розробка сайту будівельної фірми ТОВ «ІМПЕРІУМДЕВЕЛОПМЕНТ» з використанням PHP-фреймворку Laravel
Керівник Вознюк Анастасія Вадимівна

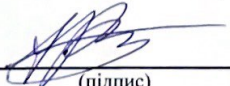
Актуальність теми: ТОВ «ІМПЕРІУМДЕВЕЛОПМЕНТ» – це будівельна фірма, яка відіграє свою роль на будівельному ринку в м. Луцьк і не тільки. Фірма пропонує послуги з зовнішніх та внутрішніх будівельних робіт. З метою покращення популяризації діяльності та інформаційного забезпечення фірми було прийнято рішення розробити сайт з використанням PHP-фреймворку Laravel.

Об'єкт дослідження – сайт ТОВ «ІМПЕРІУМДЕВЕЛОПМЕНТ». Сайт має на меті підтримку комунікації з аудиторією та просування на просторах інтернету.

Характеристика теоретичного рівня, наявності самостійних розробок і практичної значимості роботи: студентом був розроблений сайт фірми ТОВ «ІМПЕРІУМДЕВЕЛОПМЕНТ». Розробка сайту була здійснена за допомогою фреймворку Laravel, з використанням PHP та JavaScript. Сайт відповідає всім поставленим вимогам до сайту, таким як: адаптивний дизайн; просте оновлення контенту; мультимедійний контент; безпека; SEO-оптимізація. Сайт був протестований в умовах, наближених до реальних.

Зауваження та недоліки: істотних недоліків не виявлено.

Загальний висновок робота виконана на високому рівні та заслуговує позитивної оцінки за умови успішного захисту.

Керівник 
(підпис)

Вознюк Анастасія Вадимівна
(прізвище, ім'я, по батькові)

«07» червня 2024 р.

АНОТАЦІЯ

Дячук В. В. Розробка сайту будівельної фірми «ІМПЕРІУМ ДЕВЕЛОПМЕНТ» з використанням PHP-фреймворку Laravel. Рукопис.

Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2024.

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, висновків і пропозицій, списку використаних джерел.

У першому розділі здійснено аналіз розробки сайтів для будівельних компаній і були поставлені завдання до кваліфікаційної роботи. В другому розділі описано вимоги до розробки, технології, методики та алгоритми розробки і було здійснене проектування сайту. У третьому розділі розроблено сайт будівельної компанії «ІМПЕРІУМ ДЕВЕЛОПМЕНТ». У висновках узагальнено інформацію, відображену у попередніх розділах.

Ключові слова: PHP, Laravel, JavaScript, MySQL, UX, UI, SEO-оптимізація.

ABSTRACT

Dyachuk V. V. Development of the Site of the Construction Firm "Emperiumdevelopment" LLC Using the Laravel PHP Framework.

Bachelor`s Qualification Work of the Educational Program "Software Engineering". Lutsk National Technical University. Lutsk, 2024.

The bachelor`s qualification work consists of an introduction, three chapters, conclusions and proposals, and a list of references.

In the first chapter, an analysis of website development for construction companies was conducted, and the tasks for the qualification work were set. The second chapter describes the development requirements, technologies, methodologies, and development algorithms, and the website design was carried out. The third chapter developed the website for the construction company "Emperiumdevelopment". The conclusions summarize the information presented in the previous chapters.

Keywords: PHP, Laravel, JavaScript, MySQL, UX, UI, SEO optimization, adaptive design, UML diagrams.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ РОЗРОБКИ САЙТІВ ДЛЯ БУДІВЕЛЬНИХ КОМПАНІЙ І ПОСТАНОВКА ЗАВДАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ	9
1.1 Аналіз сучасного стану проблеми	9
1.2 Постанова завдання на кваліфікаційну роботу бакалавра	14
Висновки до розділу 1	14
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ.....	15
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення	15
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання	20
Висновки до розділу 2	29
РОЗДІЛ 3 РОЗРОБКА САЙТУ БУДІВЕЛЬНОЇ КОМПАНІЇ «ІМПЕРІУМДЕВЕЛОПМЕНТ».....	30
3.1 Практична реалізація об'єкта проектування	30
3.2 Тестування та налагодження інформаційно-комп'ютерної системи	39
3.3 Розробка документації для програмного продукту	40
Висновки до розділу 3	45
ВИСНОВКИ.....	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	49

ВСТУП

Актуальність теми. Розробка сучасного вебсайту для будівельної компанії ТОВ «ІМПЕРІУМДЕВЕЛОПМЕНТ» є важливим кроком у напрямку підвищення ефективності комунікації з клієнтами та покращення представлення компанії на ринку. Сайт будівельної фірми виступає як візитівка, яка не лише надає інформацію про послуги та проекти компанії, але й сприяє залученню нових клієнтів та партнерів.

З появою Інтернету та швидким розвитком цифрових технологій, наявність якісного та функціонального вебсайту стала необхідністю для будь-якої будівельної компанії. Кількість корпоративних сайтів зросла значно, і кожна компанія прагне виділитися, надаючи користувачам зручний доступ до необхідної інформації. Такі вебсайти можуть варіюватися від простих візитівок до складних платформ з інтегрованими системами управління проектами та клієнтськими базами даних.

Для клієнтів вебсайти забезпечують зручний спосіб дізнатися про послуги, переглянути портфоліо завершених проектів, отримати контактну інформацію та замовити консультацію. Крім того, такі сайти можуть надавати додаткові ресурси, такі як калькулятори вартості будівництва, онлайн-консультації та блоги з порадами.

Для будівельних компаній наявність функціонального вебсайту дозволяє розширити охоплення аудиторії, покращити обслуговування клієнтів та оптимізувати внутрішні бізнес-процеси. Сайт також може служити платформою для збору відгуків клієнтів, проведення опитувань та отримання зворотного зв'язку, що є важливим для подальшого розвитку та вдосконалення послуг.

Однак через велику кількість будівельних сайтів, клієнтам може бути важко орієнтуватися та обирати найкращі варіанти. Деякі сайти можуть бути незручними у використанні або містити застарілу інформацію.

Незважаючи на ці виклики, вебсайти залишаються ключовим інструментом у процесі залучення клієнтів та підтримки зв'язку з ними. Завдяки своїй

доступності, зручності та широкому охопленню, вебсайти надають компаніям можливість ефективно презентувати свої послуги та досягнення, сприяючи зростанню та розвитку бізнесу.

Розвиток інформаційних технологій і швидке розширення мережі Інтернет докорінно змінили спосіб ведення бізнесу в сучасному світі. Сьогодні наявність якісного вебсайту є обов'язковою умовою для успішного функціонування будь-якої компанії, незалежно від її розміру чи сфери діяльності. Це особливо актуально для будівельних фірм, які повинні забезпечувати потенційним клієнтам зручний доступ до інформації про свої послуги, реалізовані проекти та контактні дані.

Ця робота присвячена розробці вебсайту для будівельної компанії ТОВ «ІМПЕРІУМДЕВЕЛОПМЕНТ» з використанням PHP-фреймворку Laravel.

Метою даної роботи є створення сучасного, інтуїтивно зрозумілого та функціонального вебсайту, який би забезпечив ефективну комунікацію між компанією та її клієнтами, спрощував доступ до інформації про послуги та проекти, а також сприяв підвищенню рівня довіри до компанії та збільшенню її клієнтської бази.

Об'єктом даної роботи є сайт будівельної компанії ТОВ «ІМПЕРІУМДЕВЕЛОПМЕНТ».

Предметом даної роботи є практична реалізація сайту, що базується на використанні PHP-фреймворку Laravel, а також JavaScript та PHP.

Для досягнення мети дослідження були поставлені наступні завдання:

- проаналізувати актуальні проблеми та вимоги до вебсайтів;
- визначити вимоги до вебсайту та здійснити його проектування;
- обрати технології, підходи та алгоритми реалізації поставлених задач;
- здійснити практичну реалізацію об'єкта проектування;
- протестувати та відлагодити роботу розробленого вебсайту;
- розробити документацію.

РОЗДІЛ 1

АНАЛІЗ РОЗРОБКИ САЙТІВ ДЛЯ БУДІВЕЛЬНИХ КОМПАНІЙ І ПОСТАНОВКА ЗАВДАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

1.1 Аналіз сучасного стану проблеми

В сучасному світі вебсайт є основною частиною будь-якого бізнесу чи підприємства, оскільки більшість людей шукають потрібні їм послуги, в першу чергу, в інтернеті. Через це сайт є важливою частиною для розвитку та укріплення бізнесу в сучасному світі, та є сучасною та надійною заміною стереотипу пошуку різноманітних оголошень в газетах, або через знайомих, що набагато спрощує процеси залучення нових клієнтів та розповсюдження свого бренду, або послуг. На сайті можна самому контролювати подачу контенту зі своїми вимогами для нього, публікувати послуги які надає підприємство та звіти виконаних робі, що особливо привертає увагу нових клієнтів. Вебсайт залучає багато нових клієнтів та дає можливість розповсюдження підприємства в інтернеті, що в свою чергу збільшує прибуток та зменшує витрати на рекламу.

У свою чергу, сайт служить для клієнта певною гарантією надійності та відповідальності підприємства, оскільки ведення соціальних мереж швидко розповсюдилось і це може зробити кожен з власними цілями. А ось на розробку ресурсу потрібно витратити чимало часу, мати певні знання та навички, вкласти певні фінанси в розробку, що не кожен буде робити, несерйозно ставившись до своєї справи. На сайті клієнт може сам ознайомитись з всіма запропонованими підприємством послугами, дізнатись необхідну йому інформацію про підприємство, легко знайти контакти підприємства та зв'язок, ознайомитись з процесом та результатом робіт, які виконало підприємство.

Головною метою сайту – є швидке надання інформації та залучення нових клієнтів. Хоч існує багато різних способів це зробити, сайт є сучасним та ефективним способом. З його допомогою можна швидко збільшити кількість продажів, надати інформацію клієнту про послуги. Також можна добре прорекламувати послуги які надає підприємство, хоча люди ставляться до

реклами критично, в деяких випадках навіть негативно, але важко уявити сучасний світ без реклами, яка надає можливість дізнатись про існування того, чи іншого підприємства.

Сайт-візитка – це інтернет-ресурс, який містить інформацію про підприємство, його діяльність, пропоновані послуги, способи зворотного зв'язку. Сайт-візитка являє собою невеликою презентацією підприємства, містить інформацію про послуги та підприємство. Його створення є кроком для підприємства, для розширення його бізнесу та залучення нових клієнтів [1].

Для прикладу візьмемо сайт-візитку будівельної компанії «ІНВЕСТОР», при вході на який, нас зустрічає банер з написом, який відразу надає інформацію про направлення підприємства та пропоновані ним послуги. Також надається номер мобільного телефону для зв'язку, та можливість залишити заявку на зворотній зв'язок. Представлене компактне та зручне меню для навігації по сайту (рис. 1.1).

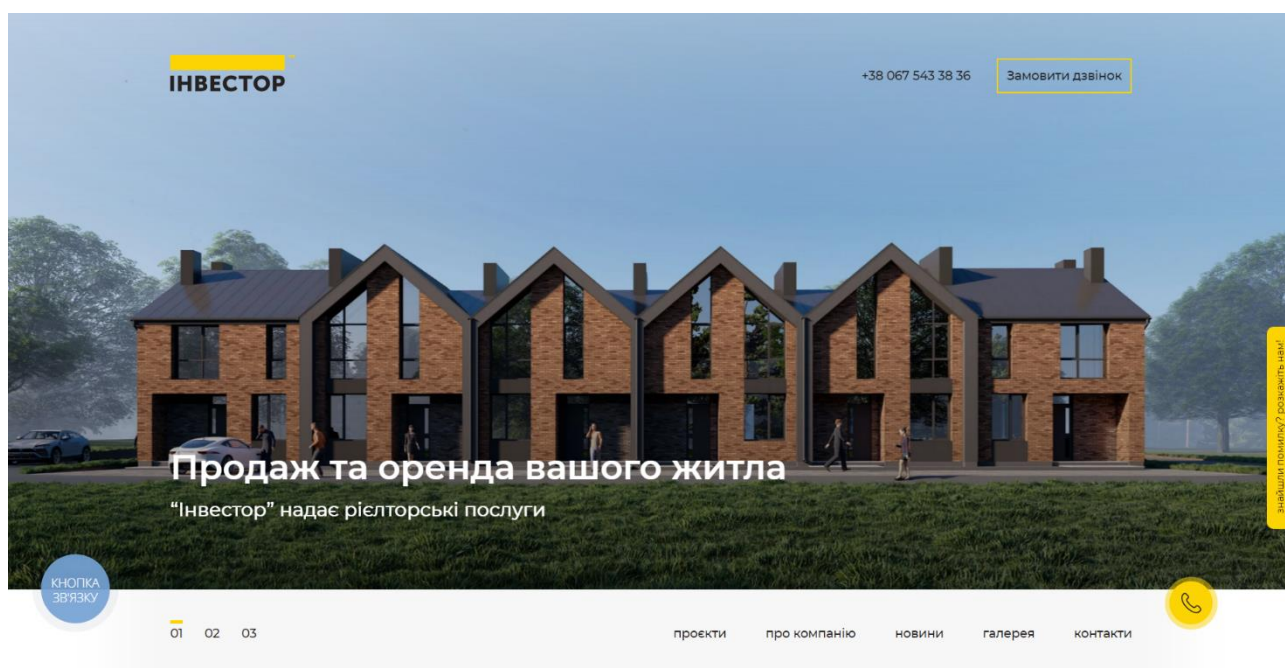


Рисунок 1.1 – Сайт-візитка будівельної компанії «ІНВЕСТОР»

На головній сторінці представлено в достатній кількості інформації, для ознайомлення новим клієнтам. А саме представлені виконані та в процесі виконання проекти,

вся потрібна до них інформація, що є зручно для відвідувачів сайту, які вперше його відвідують.

На сторінці «Про компанію» надається коротка інформація про партнерів компанії, саму компанію та її директора, що є зручно для швидкого ознайомлення відвідувачів з компанією та її діяльністю.

На сторінці «Новини» нас зустрічає стрічка з останніми новинами компанії та зручна фільтрація новин по ключовим пунктам.

На сторінці «Галерея» представлені фотозвіти виконаних компанією проектів.

На сторінці «Контакти» відвідувачу надаються контакти всіх офісів компанії та зручна форма зворотного зв'язку, в якій можна відразу залишити своє запитання (рис. 1.2).

ІНВЕСТОР +38 067 543 38 36

проекти про компанію новини галерея **контакти**

Головний офіс
 м. Луцьк, вул. Вахтангова, 12
 поштовий індекс: 43006
 (0332) 28-04-04
 office@investor.lutsk.ua

Відділ продажів бізнес-центр на Коперника
 м. Луцьк, вул. Коперника, 13 (3 поверх)
 поштовий індекс: 43025
 +38 067 543 38 36
 sale@investor.lutsk.ua

Відділ продажу ЖК «Супернова»
 м. Луцьк, вул. Супутника, 6а
 поштовий індекс: 43018
 +38 067 543 38 36
 supernova@investor.lutsk.ua

Завжди раді зустрічі з вами!
 пн-пт — 9:00 - 18:00

Ім'я: _____ Номер телефону: _____

Надіслати

знайти помилку? розкажіть нам!

Рисунок 1.2 – Сторінка «Контакти» сайту компанії «ІНВЕСТОР»

В сучасному світі існує багато РНР фреймворків для розробки сайтів, кожен з яких особливий по своєму, але зупинимось на такому фреймворці, як Laravel [2]. Laravel являється одним з популярних та розповсюджених сучасних фреймворків, який надає розробникам безліч інструментів для комфортної та

ефективної розробки сайту. Мільйони розробників з усього світу використовують цей фреймворк, що викликає довіру, оскільки велика кількість користувачів не будуть користуватись –поганим продуктом. Цей фреймворк своєю великою кількістю функцій, кількістю готових компонентів та перевагами набагато спрощує розробку (рис. 1.3).



Рисунок 1.3 – Логотип фреймворку Laravel

Розглянемо основні можливості Laravel:

- консоль Artisan – це вбудована консольна утиліта, яка надає багато команд та спрощує виконання різних завдань, дозволяє створити користувацькі команди, що дозволяє автоматизувати операції;
- конструктор Fluent надає можливість створювати та використовувати запити до бази даних без використання SQL;
- валідація. Можна використати прості правила та створити власні;
- система контролю версій баз даних. Надає засоби контролю версій баз даних, що виконується через міграції;

- юніт-тестування. Пропонує вбудовану підтримку тестування. Що надає можливість розробникам писати тести для своїх сайтів для запобігання появі помилок;

- аутентифікація. Надає зручні та прості засоби для аутентифікації користувачів.

Laravel пропонує зручний набір інструментів для розробників в створенні веб-додатків. Також цей фреймворк виділяється серед інших своїми особливостями, а саме:

- розвинена екосистема. Цей фреймворк містить багато офіційних та сторонніх пакетів, бібліотек та розширень. Що надає змогу розробникам використовувати готові рішення типових завдань;

- висока продуктивність. Ефективний та потужний движок демонструє високу продуктивність;

- відкритий код. Його код відкритий для всіх, що надає можливість розробникам вносити зміни до фреймворку;

- гнучка маршрутизація. Розробники з легкістю можуть налаштовувати та визначати дії, які повинні виконуватись при зверненні до URL;

- легка міграція. Містить в собі інструменти для керування базою даних через міграції. Що дозволяє визначити зміни бази даних та застосувати їх на середовищах розробки;

- мультимовність. Містить підтримку багатомовних сайтів, що спрощує створення сайтів з кількома мовами;

- велика кількість бібліотек та пакетів. Широка колекція бібліотек та пакетів, які можна встановити що спрощує розробникам додавання на сайт додаткових функцій розширюючи можливості фреймворку;

- регулярні оновлення. На фреймворк Laravel регулярно виходять оновлення, приблизно кожні 6 місяців, які додають нові функції, покращення продуктивності та виправлення помилок.

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

Для написання сайту-візитки потрібно дотримуватись певних правил та стандартів, щоб відвідувачу надати потрібну йому інформацію в приємному вигляді. Також потрібно акцентувати увагу на зручності сайту, оскільки відвідувач може відчувати певний дискомфорт при пошуку потрібної йому інформації.

Для виконання мети даної роботи були поставлені такі завдання:

- проаналізувати актуальні проблеми та вимоги до вебсайтів;
- визначити вимоги до вебсайту та здійснити його проектування;
- обрати технології, підходи та алгоритми реалізації поставлених задач;
- здійснити практичну реалізацію об'єкта проектування;
- протестувати та відлагодити роботу розробленого вебсайту;
- розробити документацію.

Висновки до розділу 1

Підсумовуючи, вебсайти будівельних компаній стали важливим інструментом для залучення нових клієнтів та просування послуг на ринку. Якісний веб-ресурс будівельної компанії виконує кілька ключових функцій: презентує компанію, її послуги та проекти, підвищує довіру та впізнаваність бренду, сприяє ефективній комунікації з потенційними та наявними замовниками.

Сучасний сайт будівельної компанії повинен мати інтуїтивно зрозумілий та привабливий дизайн, бути інформаційно насиченим та структурованим, оптимізованим для пошукових систем, містити інтерактивні елементи та забезпечувати безпеку даних користувачів. Дизайн сайту має бути адаптивним, відповідати фірмовому стилю компанії, а структура контенту – зручною для навігації. Важливими елементами є форми зворотного зв'язку, калькулятори вартості послуг та онлайн-консультанти.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

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

Для аналізу та визначення вимог до створення програмного забезпечення необхідно:

- розглянути функціональні характеристики вебсайтів будівельних компаній, які є основними для користувачів;
- оцінити роботу схожих вебсайтів, визначити їх можливості, переваги та недоліки;
- встановити ключові функції та особливості, які потрібно реалізувати на створюваному вебсайті будівельної компанії;
- розглянути технічні вимоги до розробки вебсайту за допомогою фреймворку Laravel, зокрема, звернути увагу на швидкодію та безпеку;
- визначити системні вимоги до розробки та необхідні інструменти і технології для успішної реалізації проєкту;
- вивчити вимоги до масштабування та розширення вебсайту в майбутньому.

З урахуванням цілей та предметної області проєкту можна виділити такі вимоги:

- реалізація головної сторінки з коротким описом компанії;
- представлення послуг компанії на окремій сторінці;
- створення сторінки контактів із зручною формою зворотного зв'язку та інтерактивною картою;
- забезпечення зручного та інтуїтивно зрозумілого інтерфейсу;
- підтримка різних браузерів та пристроїв для забезпечення доступності сайту;
- можливість інтеграції з платформами соціальних мереж для підвищення відомості про компанію та її послуги.

Можна виділити такі функціональні та не функціональні вимоги.

Функціональні вимоги:

- представлення послуг компанії на вебсайті, включаючи детальний опис та фотографії завершених проектів;
- можливість клієнтів розміщувати запити на консультацію або замовлення послуг через вебсайт;
- надання контактної інформації компанії для зв'язку з потенційними клієнтами;
- реалізація онлайн-форми для замовлення зворотного дзвінка або консультації.

Нефункціональні вимоги:

- забезпечення зручного та інтуїтивно зрозумілого інтерфейсу для користувачів;
- забезпечення швидкодії та ефективності вебсайту для швидкого доступу до інформації та замовлень;
- надійність сайту з мінімальними простоями та помилками для безперебійного доступу клієнтів;
- захист особистих даних клієнтів та їх конфіденційність;
- сумісність вебсайту з різними браузерами та пристроями для забезпечення доступності для всіх користувачів;
- можливість розширення та модернізації вебсайту для відповіді на зростаючі потреби компанії та її клієнтів.

Для створення послідовності дій потрібно визначити інтерфейси та структури даних, які забезпечать обмін інформацією між компонентами системи. У даному випадку користувач переглядає головну сторінку сайту будівельної компанії, потім переходить на сторінку із детальним описом конкретної послуги. Після ознайомлення з інформацією про цю послугу, він переходить на сторінку з контактами, де може скористатися формою зв'язку для надсилення запиту на консультацію з фахівцем компанії. Діаграма, що ілюструє взаємодію користувача з вебсайтом, зображена на рисунку 2.1



Рисунок 2.1 – Діаграма послідовностей взаємодії користувача з сайтом

Загалом, діаграма варіантів використання відтворює основну динаміку взаємодії між відвідувачем та вебсайтом будівельної компанії. Ця взаємодія охоплює функції, такі як ознайомлення зі списком послуг, пошук необхідної інформації та надання запитів на консультацію.

У діаграмі варіантів використання, зображених на рисунку 2.2, показані різноманітні сценарії взаємодії між відвідувачем та вебсайтом будівельної компанії. Особлива увага приділяється процесу пошуку інформації про послуги компанії, який ініціюється під час перебування на веб-ресурсі.



Рисунок 2.2 – Діаграма варіантів використання

Щоб створити ефективний UI/UX дизайн, необхідно врахувати наступні фактори:

Цільова аудиторія:

- аналіз користувачів. Вивчення основних категорій користувачів, їх потреб, очікувань та поведінки;
- демографічні характеристики. врахування вікових груп, статі, рівня технічної грамотності та інших демографічних показників.

Юзабіліті:

- зручна навігація. Забезпечення простої та інтуїтивно зрозумілої навігації по сайту;
- логічна організація. Організація контенту так, щоб користувачі могли легко його знайти та сприйняти.

Візуальний дизайн:

- узгодженість. Використання єдиних шрифтів, кольорів, стилів кнопок та інших візуальних елементів;
- привабливість. Створення естетично привабливого дизайну, який привертатиме увагу користувачів.

Користувацький досвід (UX):

- інтерактивні елементи. Включення елементів, які сприяють активній взаємодії користувачів із сайтом;

- зворотний зв'язок. Забезпечення можливості зворотного зв'язку, щоб користувачі могли залишати відгуки та повідомляти про проблеми.

Адаптивність:

- мобільна оптимізація. Адаптація дизайну для коректного відображення та роботи на мобільних пристроях;

- кросбраузерність. Забезпечення сумісності сайту з різними браузерами для коректного відображення.

Швидкість завантаження:

- оптимізація ресурсів. Зменшення розмірів зображень та інших ресурсів для прискорення завантаження сторінок;

- кешування. Використання кешування для зменшення навантаження на сервер та підвищення швидкості завантаження.

Доступність:

- веб-стандарти. Дотримання стандартів доступності, щоб сайт був зручним для людей з обмеженими можливостями;

- альтернативний текст. Використання альтернативного тексту для зображень та інших медіафайлів.

Контент:

- якість. Забезпечення наявності якісного, релевантного та оновлюваного контенту;

- читабельність. Використання зрозумілих та легких для читання шрифтів і структур тексту.

Безпека:

- захист даних. Забезпечення захисту даних користувачів відповідно до стандартів безпеки;

- SSL сертифікати. Використання SSL сертифікатів для захисту даних, що передаються між користувачем і сервером.

Тестування та аналітика:

- юзабіліті-тестування. Проведення регулярних тестів на юзабіліті для виявлення та виправлення проблем в інтерфейсі;
- аналіз поведінки користувачів. Використання інструментів аналітики для відстеження поведінки користувачів і вдосконалення UX.

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Вибір відповідних технологій для створення вебсайту будівельної компанії залежить конкретних вимог до нього, таких як необхідні функції та можливості сайту, вимоги до продуктивності та масштабованості, досвід і навички команди розробників у різних технологіях, а також бюджетні обмеження та строки реалізації проекту.

Ось ключові аспекти, які слід враховувати під час вибору технологічного стеку для розробки вебсайту.

Фронтенд. Відповідальний за візуальну презентацію та взаємодію з користувачем на вебсайті. Розробники зазвичай використовують HTML, CSS і JavaScript разом із сучасними фреймворками для створення динамічного та реактивного інтерфейсу. Вибір технологій може залежати від потреб у динамічному оновленні контенту та вимог до комплексності взаємодій.

Бекенд. Займається обробкою даних, логікою вебсайту та управлінням базами даних. Розробники часто вибирають між PHP, Python, Ruby, Node.js та відповідними фреймворками на кшталт Laravel, Django, Ruby on Rails чи Express.js, виходячи з потреб у швидкості обробки, безпеці та масштабованості.

База даних. Вебсайт будівельної компанії має мати змогу зберігати великий об'єм даних, таких як послуги, контент на сторінках, і запити зворотнього зв'язку.

Для керування контентом на сайті була вибрана адмін-панель Voyager. Її використання для розробки вебсайту будівельної компанії може бути обґрунтоване кількома ключовими перевагами:

- швидкість розробки. Voyager надає готове адміністративне середовище з базовими функціональностями, такими як CRUD (створення, читання, оновлення, видалення), авторизація і ролі користувачів. Це значно прискорює розробку, оскільки ви можете швидко створити адміністративний інтерфейс для управління контентом вашого сайту без необхідності писати кожен функціонал з нуля;

- гнучкість та розширюваність. Voyager дозволяє легко налаштовувати та розширювати функціональність. Ви можете додавати власні поля, створювати власні ресурси та редагувати існуючі моделі за допомогою графічного інтерфейсу. Це дозволяє адаптувати адмін панель до специфічних потреб вашого сайту будівельної компанії;

- спільнота та документація. Voyager підтримується активною спільнотою розробників, яка надає підтримку та розвиває функціонал платформи. Крім того, існує багата документація та підручники з використання Voyager, що дозволяє легко засвоїти його можливості та швидко вирішувати проблеми;

- безпека. Voyager має вбудовану систему автентифікації та авторизації, що забезпечує безпеку даних адміністративної панелі. Крім того, він має вбудовані механізми захисту від різних типів атак, таких як SQL-ін'єкції та міжсайтові атаки;

- візуальний інтерфейс. Voyager має зручний та інтуїтивно зрозумілий веб-інтерфейс, що спрощує процес управління контентом. Його структура дозволяє ефективно організувати дані та ресурси, що робить його привабливим для користувачів будь-якого рівня кваліфікації.

Підбір технологій хостингу та розгортання для вебсайту будівельної компанії має велике значення. Важливо звернути увагу на такі аспекти як здатність до балансування навантаження та автоматичне масштабування. Рішення можуть включати платформи, такі як Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform, а також Heroku та Firebase.

Безпека та продуктивність вебсайту також є критично важливими. Належне шифрування зв'язку через SSL-сертифікати, застосування сучасних методів захисту, як-от хешування паролів і перевірка коректності введення даних, сприятимуть забезпеченню надійності. Крім того, покращення швидкості вебсайту можливе завдяки використанню кешування та мережі доставки вмісту (CDN).

Врешті-решт, вибір відповідних технологій для сайту будівельної компанії має забезпечити його функціональність, високу продуктивність та надійну безпеку.

Фреймворк Laravel було обрано для розробки вебсайту будівельної компанії з кількох ключових причин.

Структурованість та чистота коду. Laravel пропонує структуроване середовище, що дозволяє розробникам писати організований і зрозумілий код. Це допомагає управляти складними проектами та сприяє легкості внесення змін і обслуговування.

Багатий набір функціоналів. Laravel включає велику кількість вбудованих можливостей, таких як аутентифікація, маршрутизація, сесії та кешування, які є невід'ємними для сучасних веб-додатків.

MVC архітектура. Фреймворк базується на архітектурі модель-вид-контролер (MVC), що забезпечує чітке розділення логіки додатку та його представлення. Це спрощує розробку та тестування.

Захист від зловмисників. Laravel має вбудовані засоби для захисту веб-додатків від загальних загроз, таких як SQL ін'єкції, XSS та CSRF, що є критично важливим для забезпечення безпеки вебсайту.

Широка спільнота та ресурси. Завдяки великій спільноті та наявності багатьох навчальних ресурсів, розробники можуть легко знаходити відповіді на свої питання та використовувати готові розширення і плагіни, що прискорює розробку.

Легкість інтеграції з іншими сервісами. Laravel легко інтегрується з різними облачними сервісами та засобами розробки, що дозволяє розширювати функціональність сайту без значних переписувань коду.

Безпека є критично важливим аспектом розробки веб-додатків, і Laravel пропонує численні вбудовані функції та інструменти для захисту вашого додатку від різних загроз. Серед основних аспектів безпеки Laravel варто виділити захист від CSRF-атак, XSS-атак, SQL-ін'єкцій, а також механізми аутентифікації та авторизації.

Захист від CSRF-атак (Cross-Site Request Forgery) забезпечується автоматичною генерацією CSRF-токенів для кожної сесії. Ці токени перевіряються під час обробки POST-запитів, що унеможливорює виконання небажаних дій від імені користувача. Для використання цієї функції достатньо додати `@csrf` у форму.

Захист від XSS-атак (Cross-Site Scripting) реалізується автоматичним екрануванням вихідних даних. Це запобігає виконанню шкідливих скриптів, введених у веб-сторінки. Використовуйте синтаксис Blade для безпечного виведення даних, наприклад, `{{ $user->name }}`.

Захист від SQL-ін'єкцій досягається використанням Eloquent ORM, який автоматично захищає від таких атак завдяки параметризованим запитам. Наприклад, запит до бази даних з використанням Eloquent буде виглядати так: `$users = DB::table('users')->where('votes', '>', 100)->get();`.

Аутентифікація та авторизація в Laravel дозволяє легко керувати доступом користувачів до різних частин додатку. Використання вбудованих функцій для реєстрації, входу в систему та відновлення паролів забезпечує надійну систему управління доступом.

Валідація вхідних даних забезпечує перевірку даних, отриманих від користувачів, на відповідність очікуваним критеріям. Це важливо для запобігання введенню некоректних або шкідливих даних.

Безпека паролів включає хешування паролів і їх перевірку. Це забезпечує безпечне зберігання паролів у базі даних та захист від витоків даних.

Використання HTTPS є критичним для безпеки додатків, особливо під час передачі чутливих даних, таких як паролі або платіжна інформація. Laravel підтримує HTTPS і може бути налаштований для використання HTTPS за замовчуванням, що можна вказати в файлі `.env`.

Ці та інші заходи допомагають забезпечити надійний рівень безпеки для додатків, побудованих на Laravel, що є невід’ємною частиною сучасної веб-розробки.

Ці фактори роблять Laravel відмінним вибором для створення надійних, масштабованих і захищених веб-додатків, що відповідають потребам сучасного бізнесу в галузі будівництва.

Webpack (рис. 2.3) є потужним інструментом для збирання модулів JavaScript, який широко використовується в Laravel через пакет Laravel Mix.



Рисунок 2.3 – Логотип Webpack

Laravel Mix надає простий API для налаштування Webpack, що значно спрощує процес компіляції та мінімізації файлів CSS і JavaScript [3].

Основні переваги використання Webpack у Laravel через Laravel Mix включають:

- простота використання. Laravel Mix спрощує складний конфігураційний процес Webpack, пропонуючи зрозумілу та інтуїтивно зрозумілу API. Це дозволяє розробникам швидко налаштовувати та виконувати задачі з компіляції активів без глибокого розуміння Webpack;
- комплексна обробка активів. Mix підтримує обробку популярних препроцесорів CSS (Sass, Less) та JavaScript (ES2015+), що робить його ідеальним вибором для сучасних веб-додатків. Завдяки цьому, можна легко компілювати та мінімізувати CSS та JS файли, що значно покращує продуктивність додатків;
- автоматизація. Використовуючи Webpack, можна автоматизувати різні аспекти розробки, такі як живий перезавантаження (hot reloading) під час розробки, що зменшує час на тестування змін;
- оптимізація продуктивності. Webpack автоматично оптимізує зібрані файли для виробничого середовища, мінімізуючи файли та видаляючи непотрібний код. Це забезпечує швидше завантаження та кращу продуктивність веб-додатків;
- модульність. Завдяки модульній структурі Webpack, можна легко інтегрувати різні модулі та бібліотеки в проєкт, що робить код більш організованим та підтримуваним.

Використання Webpack у Laravel через Laravel Mix забезпечує розробникам зручний та потужний інструмент для управління активами, спрощуючи процес розробки та покращуючи продуктивність веб-додатків.

Для розробки UI було обрано бібліотеку Bootstrap та використовувався вбудований в Laravel шаблонізатор Blade. Бібліотека Bootstrap (рис. 2.4) – це набір готових стилів і компонентів для розробки веб-інтерфейсів [4].



Рисунок 2.4 – Логотип Bootstrap

Мобільна адаптивність. Bootstrap має вбудовану підтримку мобільної адаптивності, що дозволяє автоматично адаптувати веб-сторінки до різних розмірів екранів. Це забезпечує коректне відображення вебсайту на різних пристроях, включаючи мобільні телефони і планшети.

Компоненти і шаблони. Bootstrap містить багато готових компонентів і шаблонів, таких як кнопки, форми, навігація, таблиці, картки і багато іншого. Це дозволяє розробникам швидко створювати різноманітні елементи веб-сторінок без необхідності писати власний код з нуля.

Легкість використання. Bootstrap пропонує простий та легкий у використанні синтаксис класів CSS, що дозволяє швидко додавати стилі до елементів HTML. Це дозволяє розробникам ефективно працювати з бібліотекою і зосередитися на розробці функціональності.

Щодо шаблонізатора Blade, це вбудований шаблонізатор для фреймворку Laravel. Ось кілька важливих аспектів використання шаблонізатора Blade [5].

Простота використання. Blade пропонує простий та легкий у використанні синтаксис, який дозволяє швидко створювати і керувати шаблонами веб-сторінок. Він використовує звичайний PHP-синтаксис з додатковими функціями для роботи з даними та логікою.

Вбудовані функції. Blade має вбудовані функції для роботи з даними, такі як виведення змінних, умовні оператори, цикли та інші. Це дозволяє легко вставляти динамічний контент на сторінки та керувати його відображенням.

Мастерність. Blade дозволяє розробникам використовувати мастерність, що дозволяє створювати більш складні шаблони з вкладеними структурами та повторюваними елементами, що робить процес розробки більш ефективним і зручним.

Інтеграція з Laravel. Blade ідеально поєднується з фреймворком Laravel, що забезпечує зручну роботу з функціями фреймворку, такими як маршрути, контролери і моделі. Це робить його ідеальним вибором для розробки веб-додатків на Laravel.

Swiper – це потужний і легкий інструмент для створення слайдерів на веб-сторінках, який відрізняється простотою інтеграції та широким набором функцій [6]. Використання Swiper у Laravel має численні переваги:

- простота інтеграції. Swiper легко інтегрується в Laravel проекти через npm або CDN, що дозволяє швидко додати слайдери на ваш сайт без зайвих зусиль. Це робить процес налаштування зручним і швидким;
- багатфункціональність. Swiper підтримує різноманітні функції, такі як навігаційні стрілки, пагінація, автоплей, ефекти переходів та інше. Це дозволяє створювати високофункціональні та інтерактивні слайдери, які можна легко налаштовувати під специфічні потреби проекту;
- висока продуктивність. Завдяки оптимізованому коду, Swiper працює дуже швидко і плавно навіть на мобільних пристроях. Це забезпечує відмінний користувацький досвід незалежно від пристрою, на якому відкривається сайт;
- гнучкість дизайну. Swiper дозволяє налаштовувати зовнішній вигляд слайдерів за допомогою CSS, що дає можливість створювати унікальні і стилістично відповідні елементи для вашого веб-додатку. Це забезпечує відповідність дизайну сайту загальному стилю вашого бренду;
- модульність. Swiper побудований на модульній архітектурі, що дозволяє використовувати лише ті функції, які необхідні для вашого проекту, тим самим зменшуючи розмір кінцевого коду і підвищуючи продуктивність.

Використовуючи Swiper у поєднанні з Laravel, ви можете створювати сучасні, інтерактивні та продуктивні слайдери, які легко інтегруються у ваш веб-додаток і забезпечують відмінний користувацький досвід.

MySQL (рис. 2.5) – це одна з найпопулярніших систем керування базами даних (СКБД), яка широко використовується в індустрії завдяки своїй швидкості, надійності та широкому спектру можливостей [6].



Рисунок 2.5 – Логотип MySQL

MySQL базується на реляційній моделі даних, де інформація організована у вигляді таблиць, що складаються з рядків (записів) і стовпців (полів). Ця структура дозволяє ефективно організовувати та маніпулювати даними. Для взаємодії з MySQL використовується мова структурованих запитів SQL (Structured Query Language), яка дозволяє створювати, модифікувати та витягувати дані з бази даних [7].

Серед основних переваг MySQL варто відзначити:

- висока продуктивність. MySQL забезпечує швидке виконання запитів і високу швидкість обробки даних, що робить її ідеальним вибором для веб-додатків і великих проєктів;
- надійність. MySQL відома своєю стабільністю і здатністю обробляти великі об'єми даних без втрат продуктивності;
- гнучкість налаштувань. MySQL дозволяє детально налаштовувати параметри конфігурації через файл `my.ini`, що дозволяє оптимізувати роботу сервера під конкретні потреби користувача;
- безпека. MySQL підтримує різні механізми аутентифікації та авторизації, що дозволяє забезпечити високий рівень безпеки даних.

Однією з важливих функцій MySQL є підтримка транзакцій, яка забезпечує атомарність, узгодженість, ізолюваність та довговічність (ACID) операцій з даними. Це особливо важливо для критичних систем, де необхідно гарантувати цілісність даних.

MySQL також підтримує реплікацію, що дозволяє створювати резервні копії баз даних і забезпечувати високу доступність системи. Це дозволяє масштабувати систему горизонтально, додаючи нові сервери для обробки збільшених обсягів даних.

Для взаємодії з MySQL існує безліч конекторів та драйверів, що дозволяють використовувати її з різними мовами програмування, такими як Python, PHP, Java тощо. Наприклад, популярний конектор MySQL для Python – це `mysql-connector-python`, який можна встановити через пакетний менеджер `pip` [8].

MySQL залишається популярною завдяки своїй простоті використання, широким можливостям та активному розвитку спільноти, що забезпечує постійне покращення і підтримку нових функцій та можливостей.

Висновки до розділу 2

Було ретельно досліджено специфікації для створення вебсайту, з використанням фреймворку Laravel та бібліотеки інтерфейсу користувача Bootstrap. Окреслено як функціональні, так і нефункціональні вимоги до сайту, включно з аспектами дизайну UX/UI, які мають бути враховані. Подальше створення різних UML діаграм, включаючи діаграму варіантів використання, діаграму послідовності та діаграму стану, дозволило деталізувати вимоги та глибше зрозуміти поведінку системи. Також поставлено завдання для наступних етапів розробки.

Крім того, було акцентовано на важливості обрання оптимальних технологій для проєкту, і виділено ключові переваги застосування Laravel та Bootstrap. Laravel було вибрано за його здатність до швидкої розробки, масштабованість та інтегрованість, в той час як Bootstrap використано через його ефективність у створенні адаптивних і естетичних користувацьких інтерфейсів.

РОЗДІЛ 3

РОЗРОБКА САЙТУ БУДІВЕЛЬНОЇ КОМПАНІЇ «ІМПЕРІУМДЕВЕЛОПМЕНТ»

3.1 Практична реалізація об'єкта проєктування

Отже, основними вимогами до сайту-візитки є:

- привабливий дизайн, який повинен мати мінімалістичний та сучасний вигляд, щоб приваблювати нових відвідувачів;
- інформація про компанію. На сайті повинен бути короткий опис компанії, її мети та цілей;
- контактна інформація. На сайті повинні бути номер телефону, електронна пошта, адреса або місцезнаходження офісу, форма зворотного зв'язку;
- фотографії та мультимедіа. На сайті повинні бути відеозвіти, або фото виконаних робіт, для ознайомлення відвідувачами;
- адаптивний сайт. Сайт повинен бути адаптований та коректно відображатись на мобільних телефонах, планшетах, комп'ютерах;
- соціальні мережі. На сайті повинні бути зручні посилання на сторінки в соціальних мережах.

Для реалізації вебсайту будівельної компанії потрібно для початку створити проєкт Laravel. Встановивши усі необхідні технології розробки, а саме PHP, Composer та сервер БД MySQL та створивши директорію, її було відкрито у редакторі коду та в його терміналі була виконана команда `'composer create-project --prefer-dist laravel/laravel myproject'`, після чого встановився фреймворк Laravel [9]. Для його роботи також потрібно створити базу даних та налаштувати оточення. У лістингу 3.1 описана конфігурація бази даних у файлі `.env` мого проєкту.

Лістинг 3.1 – Конфігурація бази даних

`DB_CONNECTION=mysql`

```
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=estate
DB_USERNAME=root
DB_PASSWORD=root
```

Кінець лістингу 3.1

Коли проект Laravel функціонував коректно, було встановлено адмін-панель Voyager. Для початку додали пакет Voyager за допомогою Composer. Було виконано команду ‘composer require tcg/voyager’, що завантажило та встановило пакет Voyager [10]. Після завершення встановлення було необхідно додати провайдера служби та фасад у файл config/app.php. Додавання було здійснено наступним чином: у секцію providers додано рядок TCG\Voyager\VoyagerServiceProvider::class, а у секцію aliases - рядок «Voyager»=> TCG\Voyager\Facades\Voyager::class.

Наступним кроком було створення таблиць бази даних, необхідних для роботи Voyager (рис. 3.1). Для цього була виконана команда php artisan voyager:install, яка мігрувала базу даних та встановила всі необхідні компоненти. Під час цього процесу було запропоновано створити адміністратора. Після завершення команди виведено повідомлення про успішне встановлення.



Рисунок 3.1 – Логотип Voyager

Щоб забезпечити доступ до адмін-панелі, створено користувача-адміністратора. Виконано команду «php artisan voyager:admin admin@admin.com --create», яка створила нового адміністратора з електронною поштою admin@admin.com. Під час виконання команди було надано інструкції для

встановлення пароля та завершення налаштувань адміністратора. Тепер адмін-панель встановлена та функціонує правильно (рис. 3.2).

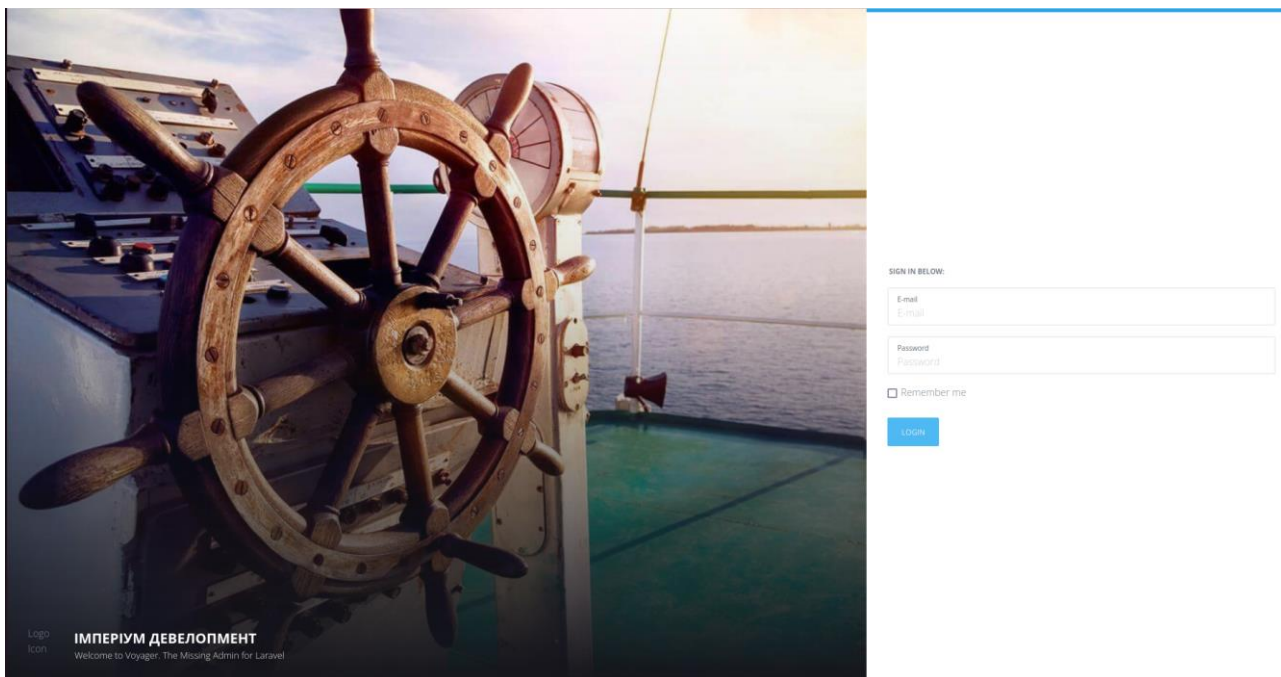


Рисунок 3.2 – Вигляд сторінки входу адмін-панелі

Спершу було створено функціонал послуг. Для цього потрібно було в адмін-панелі створити таблицю services (рис. 3.3).

Editing table services

Table Name
services

Table Columns

Name	Type	Length	Not Null	Unsigned	Auto Increment	Index	Default	
id	INTEGER		☒	☒	☒	PRIMARY		
title	TEXT	65535	☐	☐	☐			
content	LONGTEXT	0	☐	☐	☐			
main_image	TEXT	65535	☐	☐	☐			
created_at	TIMESTAMP	0	☐	☐	☐			
updated_at	TIMESTAMP	0	☐	☐	☐			
subTitle	TEXT	65535	☐	☐	☐			

[+ Add New Column](#)
[+ Add Timestamps](#)
[+ Add Soft Deletes](#)

[Update Table](#)

Рисунок 3.3 – Налаштування таблиці Services

Після цього потрібно налаштувати CRUD-операції з таблицею (рис. 3.4). Наступним кроком було створення моделі цієї таблиці.

Field	Visibility	Input Type	Display Name	Optional Details
+ id Type: integer Key: PK Required: Yes	☐ Browse ☐ Read ☐ Edit ☐ Add ☐ Delete	Text	id	
+ title Type: text Key: Required: No	☒ Browse ☒ Read ☒ Edit ☒ Add ☒ Delete	Text	Title	
+ subtitle Type: text Key: Required: No	☒ Browse ☒ Read ☒ Edit ☒ Add ☒ Delete	Text	Subtitle	
+ content Type: longtext Key: Required: No	☒ Browse ☒ Read ☒ Edit ☒ Add ☒ Delete	Rich Text Box	Content	
+ main_image Type: text Key: Required: No	☒ Browse ☒ Read ☒ Edit ☒ Add ☒ Delete	Image	Main Image	
+ created_at Type: timestamp Key: Required: No	☒ Browse ☒ Read ☒ Edit ☐ Add ☒ Delete	Timestamp	Created At	
+ updated_at Type: timestamp Key: Required: No	☐ Browse ☐ Read ☐ Edit ☐ Add	Timestamp	Updated At	

Рисунок 3.4 – Налаштування CRUD таблиці Services

Після цього було створено Blade-шаблон головної сторінки, контроллер для неї та маршрут до головної сторінки (ліст. 3.2).

Лістинг 3.2 – Маршрут головної сторінки

```
Route::get('/', [HomeController::class, 'index'])->name('home');
```

Кінець лістингу 3.2

Згодом було розроблено фронтент-частину (ліст. 3.3).

Лістинг 3.3 – Реалізація фронтенду для блоку послуг

```
<div class="row d-flex container flex-wrap services__container">
  @foreach($services as $service )
    <div class="col-md-6 col-12" style="padding: 10px" >
      <div class="card bg-dark text-white">
        
        <div class="card-img-overlay">
```

```

<h5 class="card-
title">{{$service->title}}</h5>
<p class="card-
text">{{$service->subtitle}}</p>
<a href="service/{{$service->id}}"
class="my-button">Дізнатись більше</a>
</div>
</div>
</div>
@endforeach
</div>

```

Кінець лістингу 3.3

Аналогічно було розроблено інші блоки на головній сторінці. Наступним кроком була розробка меню. Для цього у адмін-панелі Voyager є конструктор меню (рис. 3.5).

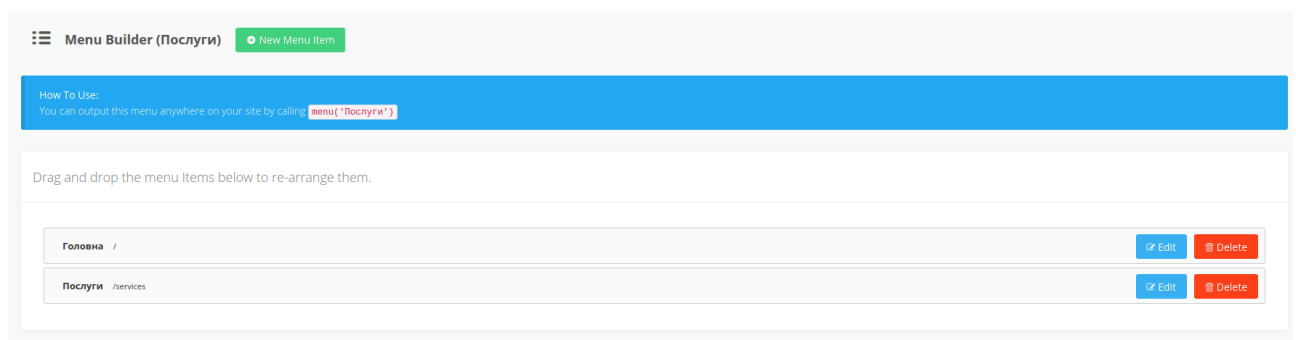


Рисунок 3.5 – Конструктор меню

Для того, щоб меню з'явилося на сторінці потрібно передати об'єкт меню у шаблон. Замість того, щоб використовувати маршрут, я передав об'єкт меню у всі наявні шаблони з допомогою методу `boot()` провайдера `AppServiceProvider` (ліст. 3.4).

Лістинг 3.4 – Метод `boot()` провайдера `AppServiceProvider`

```

public function boot(): void {
    view()->composer('*', function ($view) {
        $view->with('menus', MenuItem::where('menu_id', '2')->get());
    }); }

```

Кінець лістингу 3.4

Після цього розробка головної сторінки була завершена. Вигляд головної сторінки зображено на рисунку 3.6.

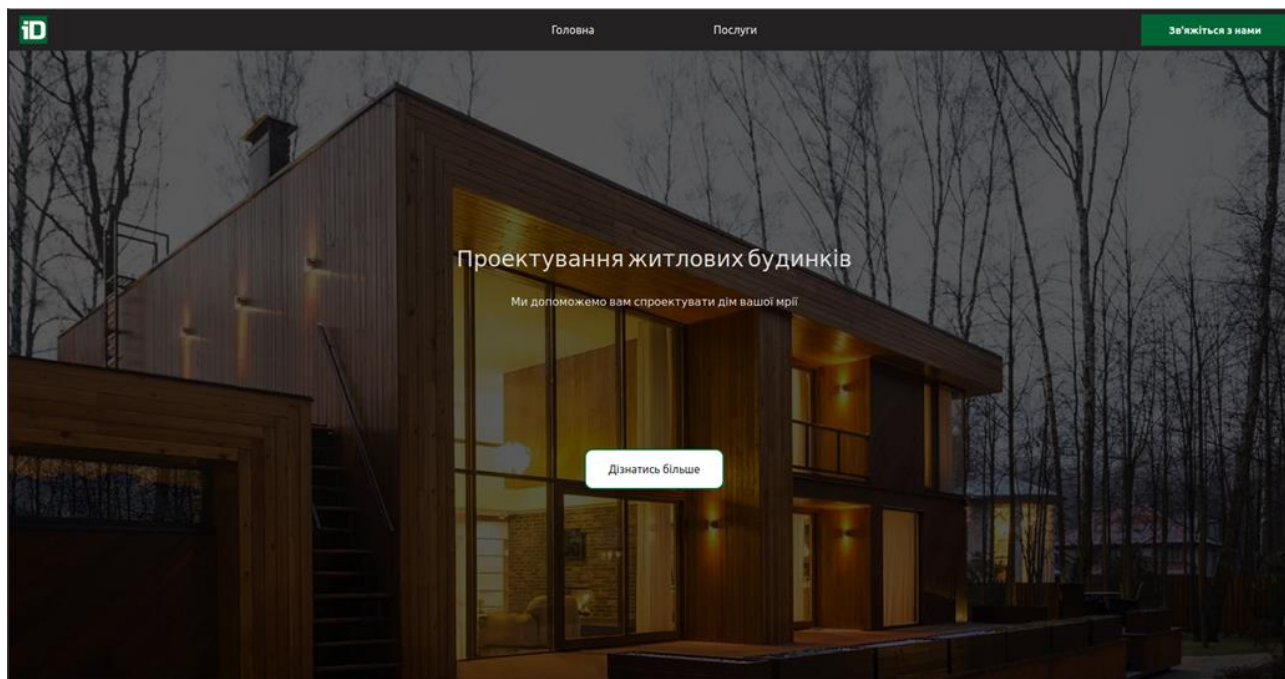


Рисунок 3.6 – Головна сторінка

Схожим чином була реалізована сторінка послуг (рис. 3.7).

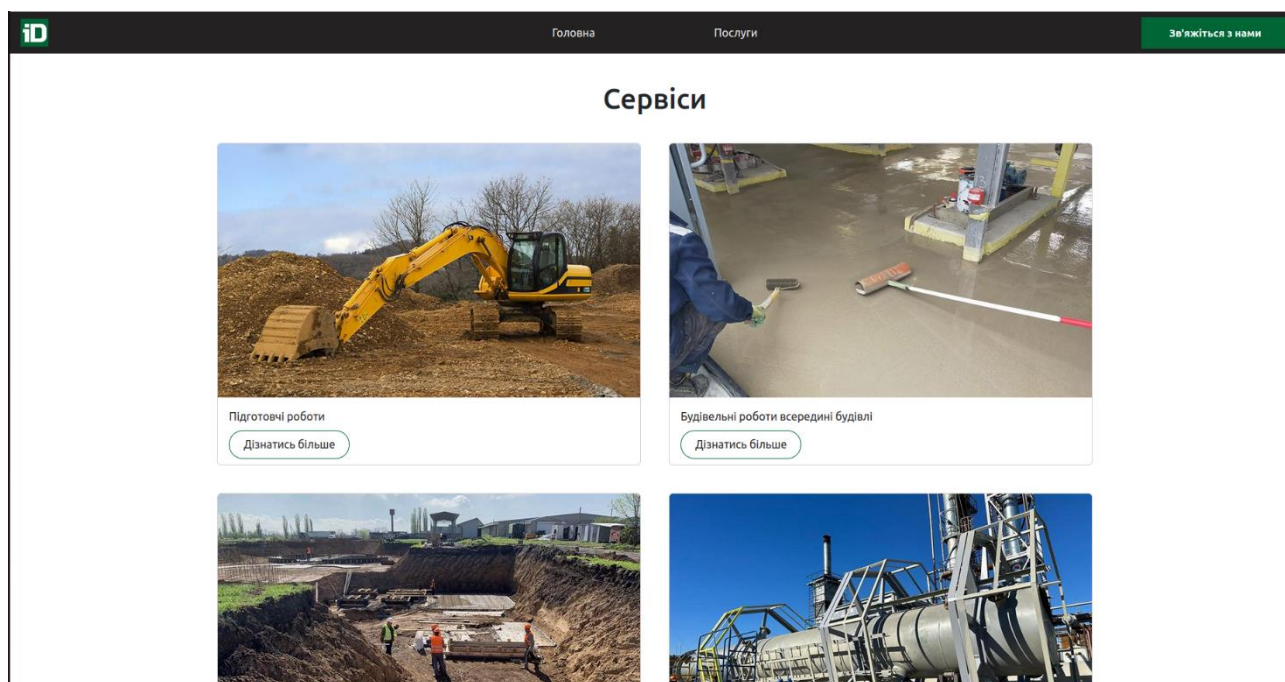


Рисунок 3.7 – Сторінка «Послуги»

Згодом я реалізував зворотній зв'язок. Спершу потрібно було знову змінити налаштування оточення (ліст. 3.5). В якості поштового сервера я використав Gmail SMTP [11].

Лістинг 3.5 – Налаштування поштового сервера

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.gmail.com
MAIL_PORT=587
MAIL_USERNAME=imperiumdevelopmentcommerce@gmail.com
MAIL_PASSWORD=example
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS="imperiumdevelopmentcommerce@gmail.com"
MAIL_FROM_NAME="Імперіум Девелопмент"
```

Кінець лістингу 3.5

Після налаштування оточення потрібно було зробити контроллер, котрий буде приймати данні з форми та будувати лист (ліст. 3.6)

Лістинг 3.6 – Контроллер створення листа

```
class ContactFormMail extends Mailable
{
    use Queueable, SerializesModels;

    public $details;

    public function __construct($details)
    {
        $this->details = $details;
    }

    public function build()
    {
        return $this->subject('no-reply')
        ->view('emails.contact');
    }
}
```

Кінець лістингу 3.6

Після цього потрібно створити шаблон листа, код шаблона представлено у лістингу 3.7

Лістинг 3.7 – Шаблон листа

```
<p>Доброго дня! Ми отримали ваш запит. Наш спеціаліст скоро
зв'яжеться з вами. Ось деталі вашого звернення:</p>
<p>Ваше ім'я: {{ $details['name'] }}</p>
<p>Ваш Email: {{ $details['email'] }}</p>
<p>Ваше питання: {{ $details['message'] }}</p>
<p>Очікуйте на дзвінок.</p>
<p>З повагою, команда Імперіум девелопмент!</p>
```

Кінець лістингу 3.7

Реалізувавши функціонал, потрібно було створити сторінку «Контакти» та форму зворотнього зв'язу, створити контроллер (ліст. 3.8).

Лістинг 3.8 – Файл ContactController.php

```
class ContactController extends Controller
{
    public function showForm()
    {
        return view('contact');
    }

    public function sendEmail(Request $request)
    {
        $request->validate([
            'name' => 'required',
            'email' => 'required|email',
            'message' => 'required',
        ]);

        $details = [
            'name' => $request->name,
            'email' => $request->email,
            'message' => $request->message,
        ];

        Mail::to( $request->email)->send(new ContactFormMail($details));
```

```
return back()->with('message', 'Your message has been sent!');
}
}
```

Кінець лістингу 3.8

Та додати маршрути сторінки контактів (ліст. 3.9).

Лістинг 3.9 – Маршрути сторінки «Контакти»

```
Route::get('/contact', [ContactController::class,
'showForm'])->name('contact');
Route::post('/contact', [ContactController::class,
'sendEmail'])->name('contact.send');
```

Кінець лістингу 3.9

Реалізувавши фронтенд-частину сторінка набула такого виду, як на рисунку 3.7.

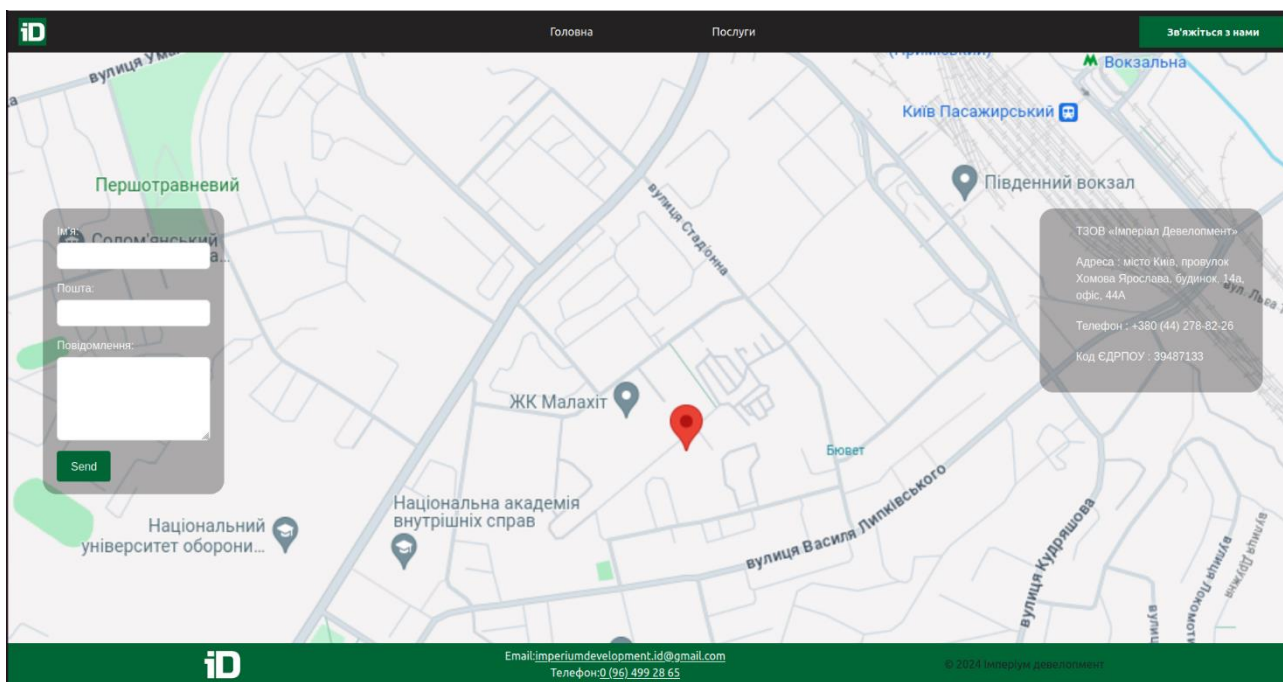


Рисунок 3.7– Сторінка «Контакти»

3.2 Тестування та налагодження інформаційно-комп'ютерної системи

Було виконане тестування та налаштування інформаційно-комп'ютерної системи для вебсайту будівельної компанії з використанням Laravel, що дало цінний досвід у процесах тестування та налагодження. Спочатку необхідно наголосити на значущості тестування, яке є критичним етапом під час розробки будь-якого програмного забезпечення.

Тестування є ключовим аспектом у розробці програмного забезпечення, забезпечуючи його якість, функціональність та надійність. Для вебсайту будівельної компанії, ретельне тестування мало критичне значення для забезпечення ідеальної взаємодії з користувачами, перевірки впроваджених функцій, а також виявлення та усунення будь-яких збоїв або дефектів. Це допомогло покращити загальну продуктивність і користувацьку зручність вебсайту [12].

При розробці вебсайту я використовував кілька методів тестування для забезпечення його надійності та ефективності. Ось основні способи.

Модульне тестування (Unit Testing). Цей метод дозволив мені перевірити окремі компоненти сайту на відповідність зазначеним функціональним вимогам. Зокрема, кожен клас та метод був протестований ізольовано від інших частин системи. Це допомогло виявити і виправити базові помилки на ранніх стадіях розробки.

Інтеграційне тестування (Integration Testing). Після модульного тестування я перейшов до інтеграційного, яке фокусувалось на взаємодії між різними модулями та компонентами системи. Це допомогло забезпечити, що всі частини вебсайту працюють разом гармонійно, без помилок інтеграції.

Функціональне тестування (Functional Testing). Цей тип тестування був використаний для перевірки, чи вебсайт в цілому виконує всі необхідні функції згідно з вимогами користувачів і специфікаціями. Тестування охоплювало різні сценарії використання сайту, відповідно до його призначення.

Навантажувальне тестування (Load Testing). Цей спосіб дозволив визначити, як сайт веде себе під великим навантаженням, наприклад, коли до нього одночасно звертається багато користувачів. Це важливо для гарантування того, що вебсайт може стабільно працювати під час піків відвідуваності.

Захисне тестування (Security Testing). Я переконався, що сайт захищений від зовнішніх загроз та атак. Це включало тестування на вразливості, такі як SQL ін'єкції, XSS (Cross-site scripting) і CSRF (Cross-Site Request Forgery).

Кожен з цих методів тестування допоміг мені підвищити якість продукту, знизити кількість помилок у роботі сайту і забезпечити хороший досвід для кінцевих користувачів.

Розробка та налагодження вебсайту будівельної компанії на базі Laravel включали ключові етапи, спрямовані на забезпечення його ефективності, стабільності та зручності використання. Застосування ретельних методів тестування, таких як модульне, інтеграційне та наскрізне, дозволило виявити та усунути помилки. Аналізуючи роботу, були визначені недоліки та внесені відповідні покращення. Крім цього, для діагностики проблем у системі та їх вирішення застосовувалися ефективні методи налагодження, такі як аналіз журналу консолі, огляд коду та використання інструментів налагодження. Цей досвід вклав цінну інформацію щодо важливості тестування та налагодження у процесі розробки програмного забезпечення, а також надав мені необхідні навички для створення високоякісних та надійних веб-додатків у майбутньому.

3.3 Розробка документації для програмного продукту

Мінімальні системні вимоги.

Операційна система: Windows 10 і новіші версії, macOS 10.10 і новіші версії, Linux (підтримуються дистрибутиви на кшталт Ubuntu, Fedora).

Інтернет-браузер: останні версії Google Chrome, Mozilla Firefox, Safari, Microsoft Edge.

Інтернет: швидкісне підключення для найкращої продуктивності.

Керівництво користувача для вебсайту будівельної компанії, яке детально охоплює всі його функції та можливості. Посібник включає покрокові інструкції щодо відправки контактної форми.

Відправлення форми зворотного зв'язку:

- дочекайтеся підтвердження про успішне відправлення форми, яке з'явиться на екрані або буде надіслане на вашу електронну пошту;
- відвідайте домашню сторінку вебсайту;
- заповніть форму зворотного зв'язку, вказавши своє ім'я, адресу електронної пошти, та сам текст повідомлення;
- перевірте правильність введеної інформації;
- натисніть кнопку «Відправити».

Додавання нової послуги.

Увійдіть в адмін-панель за посиланням /admin. При переході відобразиться сторінка входу (рис. 3.8).

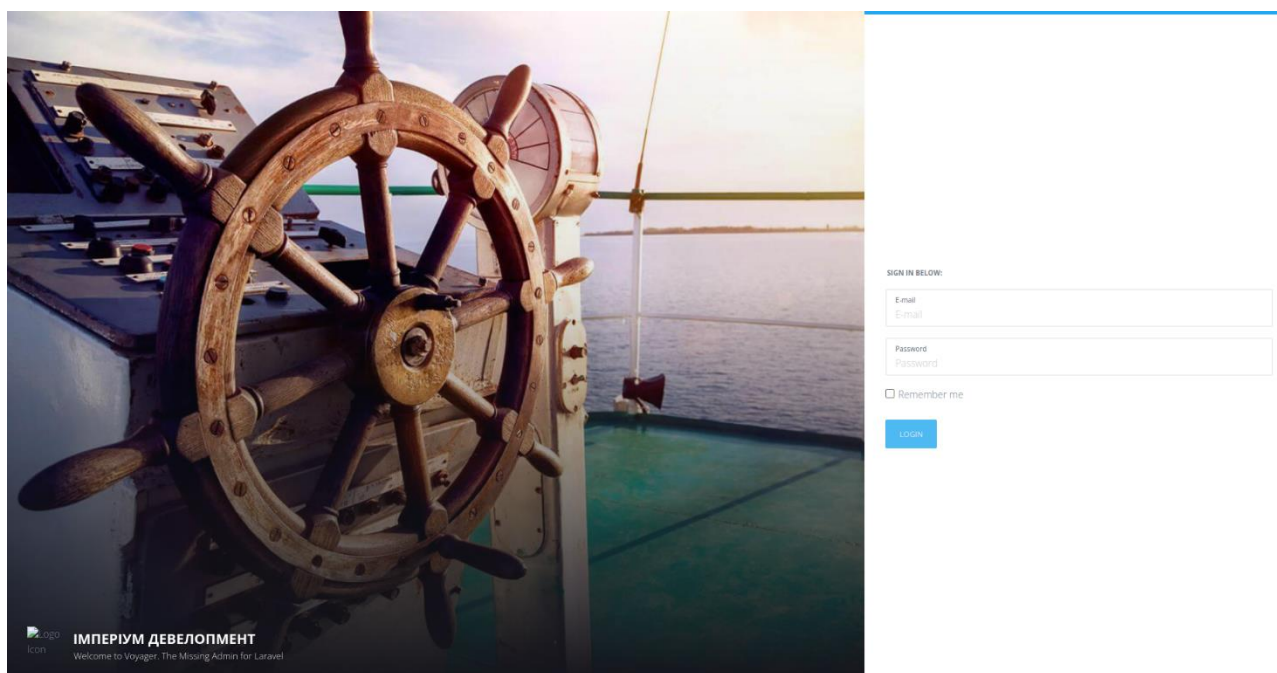


Рисунок 3.8 – Адмін-панель Voyager

Потрібно ввести відповідні данні у поля на натиснути кнопку входу. Після цього відобразиться адмін панель на вкладці інформаційної панелі (рис. 3.9).

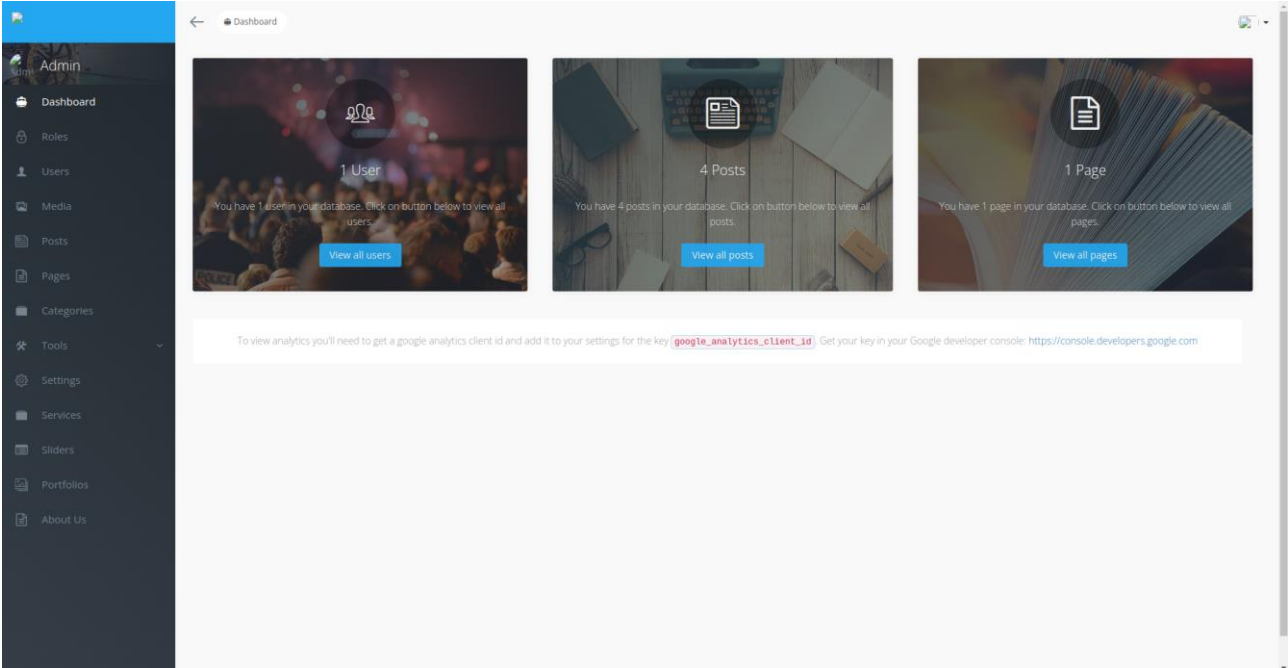


Рисунок 3.9 – Інформаційна панель

На меню, розташованому зліва, потрібно обрати пункт Services та клацнути по ньому. Після цього адміністратора перенаправить на сторінку, на якій будуть відображені всі сервіси (рис. 3.10).

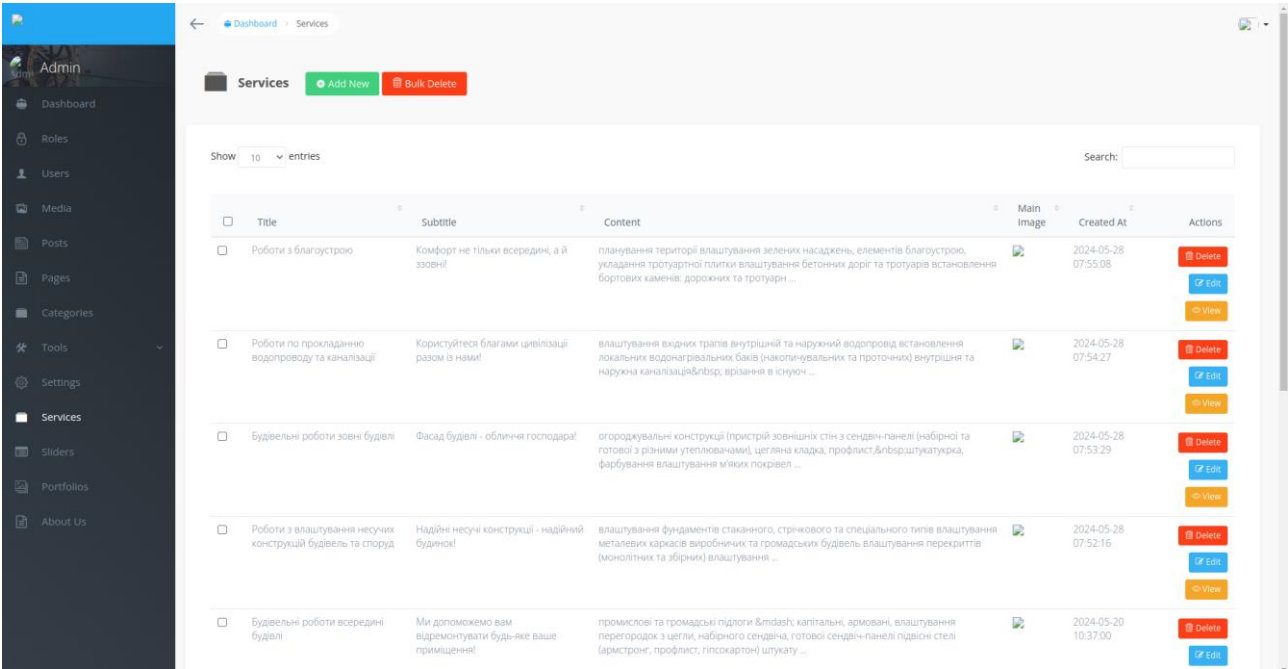


Рисунок 3.10 – Сервіси

Наступним кроком буде натискання кнопки додавання нового сервісу, що розташована зверху. Адміністратора перенаправить на форму створення нового сервісу (рис. 3.11).

Рисунок 3.11 – Форма додавання сервісу

Після вводу інформації та натискання кнопки збереження послуга буде додана на сайт, адміністратора перенаправить на сторінку, на якій відображаються всі сервіси. Схожим чином додаються усі інші елементи на сайт, такі як слайди, елементи портфоліо або інформація з блоку, в якому описується компанія. Для редагування слід на сторінці сервісів натиснути відповідну кнопку, яка розташована з правого краю кожного окремого блоку з короткою інформацією про сервіс. Так само працює і видалення.

Редагування меню.

Для того, щоб відредагувати меню, потрібно виконати перший пункт попереднього сценарію, щоб опинитись на сторінці інформаційної панелі. Згодом потрібно обрати пункт Tools та у випадавшому списку обрати пункт Menu Builder. Виконавши це, адміністратора перенаправить на сторінку, на якій відображено всі доступні меню – меню адмін-панелі та меню сайту (рис. 3.12).

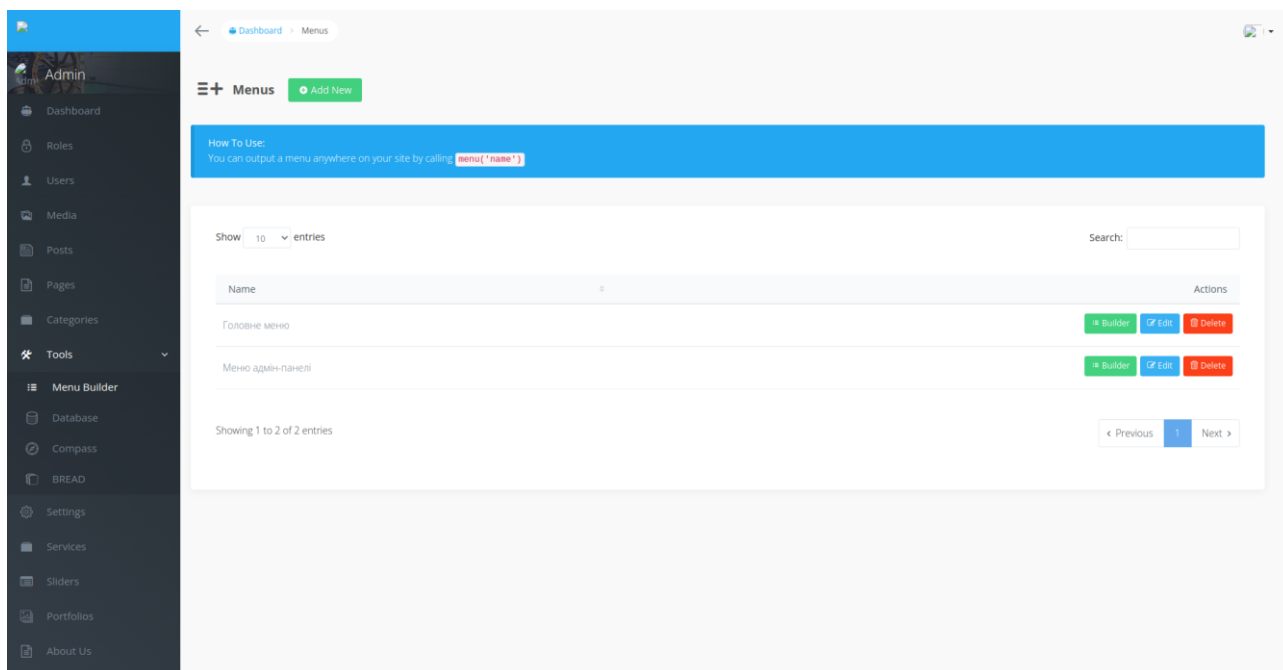


Рисунок 3.12 – Сторінка редагування меню

Обравши потрібне вам меню натисніть на кнопку конструктора у блоці відповідного меню. Після цього адміністратора перенаправить на сторінку редагування (рис. 3.13).

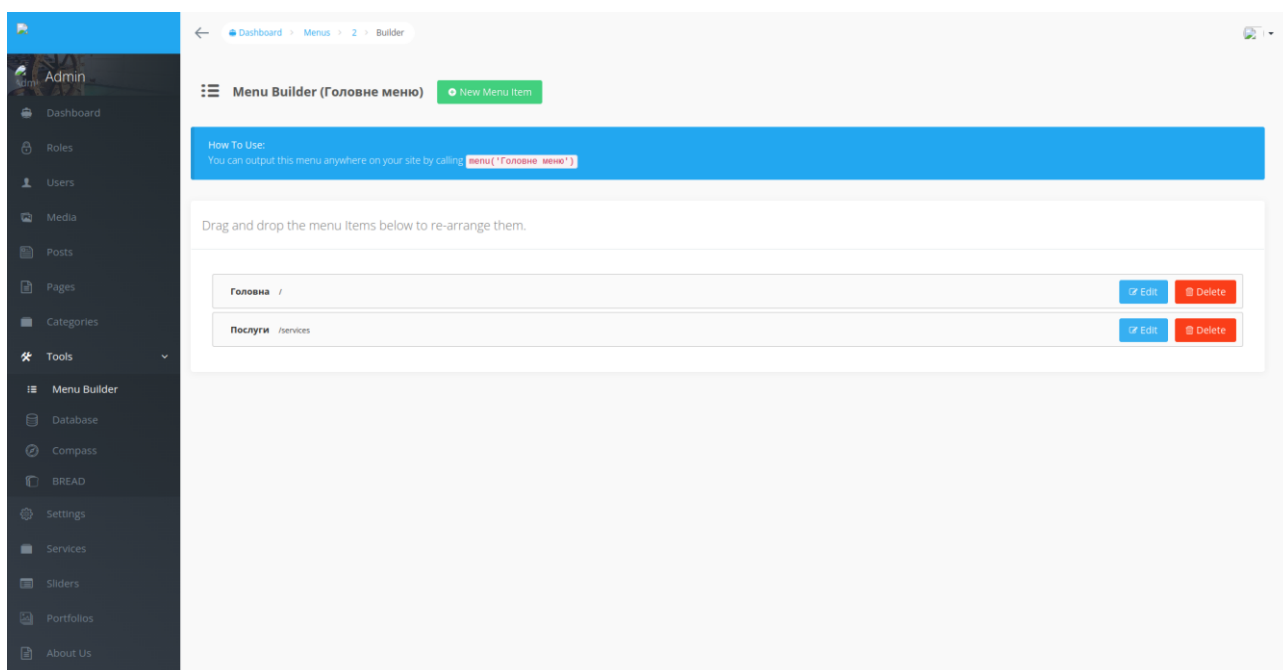


Рисунок 3.13 – Сторінка редагування окремого меню

Для додавання потрібно натиснути відповідну кнопку вверху сторінки, після чого адміністратора перенаправить на форму, де потрібно ввести відповідні данні та натиснути кнопку збереження. Схожим чином працює і редагування та видалення елемента меню.

Висновки до розділу 3

В цьому розділі було описано процес створення проекту, встановлення адмін-панелі, розробка сторінок та форми зворотнього зв'язку.

Розроблений сайт був протестований для всіх можливих сценаріїв використання, було виправлено деякі виявлені несправності.

Також було розроблено посібник користувача з покроковою інструкцією як користуватися сайтом.

ВИСНОВКИ

Розробка вебсайту будівельної компанії була успішно виконана відповідно до поставлених завдань. Сайт надає користувачам легкий та зручний доступ до інформації про послуги компанії.

Також користувачам надається можливість подавати запити на консультації.

Поставлені завдання були виконані. Сайт розроблений таким чином, що він є масштабованим та легко розширюваним, що дозволяє компанії вдосконалювати платформу відповідно до зростаючих потреб клієнтів.

Адаптивний дизайн сайту гарантує зручний перегляд і використання на різних пристроях, забезпечуючи максимальну сумісність і комфорт для користувачів. Дизайн коректно функціонує на мобільних пристроях, планшетах, а також на ПК з розширенням 4K. Сайт стабільно працює на попередніх версіях браузерів. Медіа адаптовані під Retina дисплеї.

Важливим аспектом є забезпечення безпеки даних користувачів. Розроблена система включає захист від хакерських атак та незаконного доступу до даних, забезпечуючи конфіденційність та цілісність інформації. Використана технологія Firebase забезпечує максимальний захист особистих даних користувачів.

Створені зручні умови для комунікації між клієнтом та компанією, що покращує потенційну співпрацю.

Загалом, розроблений сайт будівельної компанії відповідає всім поставленим завданням, забезпечуючи зручний, швидкий та безпечний доступ до інформації про послуги компанії, сприяючи успішному розвитку співпраці з клієнтами.

Під час роботи над кваліфікаційною роботою бакалавра були виконані усі поставлені задачі.

Першим кроком стало дослідження поточних тенденцій та вимог до вебсайтів. Це дало змогу визначити, які функції та можливості повинні бути реалізовані у розроблюваному сайті, щоб відповідати очікуванням користувачів та замовників. Також був проведений аналіз існуючого дизайну сайту, що допомогло виявити його переваги та недоліки.

Наступним етапом було визначення вимог до розроблюваного сайту та його проектування. Було визначено, що сайт повинен відповідати наступним критеріям:

- мати адаптивний дизайн для зручного перегляду на різних пристроях;
- використовувати Laravel для полегшення процесу оновлення контенту;
- підтримувати відображення мультимедійних матеріалів, таких як зображення, відео та аудіо;
- забезпечити високий рівень безпеки даних завдяки можливостям Laravel;
- бути оптимізованим для пошукових систем для кращої видимості у результатах пошуку.

Ці вимоги сформували основу для проектування та розробки майбутнього сайту.

Важливо, щоб сайт був легким у використанні та інтуїтивно зрозумілим для користувачів.

На цьому етапі було розроблено діаграми, які ілюструють послідовність взаємодій користувача з сайтом, а також варіанти використання. Діаграми включають актора «Користувач» та описують різні сценарії взаємодії з сайтом.

На етапі вибору технологій та методів вирішення завдання було вирішено використовувати платформу Laravel для розробки сайту з використанням JavaScript і PHP. Для адміністрування контенту було обрано Laravel у поєднанні з Voyager, що забезпечує зручний інтерфейс для управління сайтом. Front-end ча-

стина буде реалізована на JavaScript, а back-end частина - на PHP. Також були визначені ефективні методи SEO-оптимізації для покращення позицій сайту в пошукових системах.

Була здійснена практична імплементація проекту, де проводилися верстка та програмування сайту. Front-end частина, що відповідає за зовнішній вигляд і взаємодію з користувачем, та back-end частина, яка відповідає за логіку та з'єднання з базою даних, були реалізовані. Коли сайт був наповнений контентом, він пройшов тестування на помилки та функціональність. Були виправлені всі виявлені недоліки, а також проведена оптимізація для поліпшення швидкості завантаження.

Останнім кроком є SEO-оптимізація сайту, яка включає ряд заходів з метою підвищення його видимості в пошукових системах. На цьому етапі проводиться оптимізація контенту сайту під ключові запити, налаштовуються мета-теги та заголовки сторінок, а також будується зв'язки з інших ресурсів.

Крім того, було розроблено детальну документацію користувача з покроковою інструкцією щодо користування сайтом. Ця документація включає в себе пояснення основних функцій, поради з налаштування профілю, використання інтерактивних елементів та вирішення можливих проблем.

Після втілення поставлених цілей, сайт будівельної компанії «ІМПЕРІУМДЕВЕЛОПМЕНТ» був успішно розроблений. Цей сайт став зручним інструментом для підтримки бренду та взаємодії з потенційними клієнтами. Він забезпечує швидкий доступ до інформації про компанію, її послуги та портфоліо проектів. Сайт допомагає клієнтам отримати детальну інформацію про діяльність компанії, оцінити її професійність на основі раніше реалізованих проектів та зв'язатися з її представниками для отримання консультації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Що таке сайт-візитка і яким він повинен бути | Створити сайт в web-студії WebKitchen %. Створити сайт в web-студії WebKitchen. URL: <https://webkitchen.kiev.ua/ua/blog-ua/shho-take-sajt-vizitka-i-yakim-vin-povinen-buti> (дата звернення: 14.03.2024).
2. Laravel – The PHP Framework For Web Artisans. Laravel – The PHP Framework For Web Artisans. URL: <https://laravel.com/docs/11.x/mix> (дата звернення: 04.04.2024).
3. Laravel Mix. Laravel Mix. URL: <https://laravel-mix.com/docs/6.0/installation> (дата звернення: 04.04.2024).
4. Bootstrap 3 Підручник. Старт. Уроки для початківців. W3Schools українською. W3Schools українською. Безплатні уроки онлайн для початківців, школярів та студентів. URL: https://w3schoolsua.github.io/bootstrap/bootstrap_get_started.html#gsc.tab=0 (дата звернення: 07.04.2024).
5. Understanding Laravel Blade and How To Use It. Kinsta®. URL: <https://kinsta.com/blog/laravel-blade/> (дата звернення: 15.04.2024).
6. Swiper – the most modern mobile touch slider. Swiper. URL: <https://swiperjs.com/> (дата звернення: 17.04.2024).
7. Відеокурс MySQL базовий для початківців українською мовою - відеоуроки програмування | ITVDN. ITVDN. URL: <https://itvdn.com/ua/video/mysql-essential-ua> (дата звернення: 23.04.2024).
8. Guevara H. Build a Laravel application with a MySQL database – PlanetScale. The ultimate MySQL database platform – PlanetScale. URL: <https://planetscale.com/blog/build-a-laravel-application-with-a-mysql-database> (дата звернення: 23.04.2024).
9. PHP + MySQL. URL: <http://sites.znu.edu.ua/webprog/lect/1222.ukr.html> (дата звернення: 27.04.2024).
10. Laravel Voyager. Voyager – The Missing Laravel Admin. URL: <https://voyager.devdojo.com> (дата звернення: 28.04.2024).

11. IMAP, POP, and SMTP | Gmail | Google for Developers. Google for Developers. URL: <https://developers.google.com/gmail/imap/imap-smtp> (дата звернення: 05.05.2024).

12. Hu E. How to Get Started with Unit Testing in Laravel | Better Stack Community. Better Stack: Spot, Resolve, and Prevent Downtime. URL: <https://betterstack.com/community/guides/testing/laravel-unit-testing/> (дата звернення: 15.05.2024).