

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-ВІЗИТКИ ДЛЯ КОМПАНІЇ З
МІЖНАРОДНИХ ПЕРЕВЕЗЕНЬ «MPL»

DEVELOPMENT OF A BUSINESS CARD WEBSITE FOR THE
INTERNATIONAL TRANSPORTATION COMPANY "MPL"

спеціальність 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

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

Виконав: здобувач вищої освіти
Групи ПЗ-41
Гузоватий Сергій Володимирович


(підпис)

Керівник:
д.т.н., професор
Андрушак Ігор Євгенович


(підпис)

Кваліфікаційну роботу
допущено до захисту
«5» 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

М.А. Андрієв
« 2 » 02 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Гузоватому Сергію Володимировичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Розробка сайту-візитки для компанії з міжнародних перевезень «MPL»

Керівник роботи: д.т.н., професор, Андрущак Ігор Євгенович

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

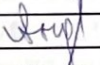
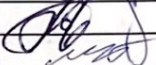
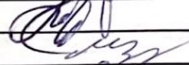
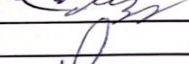
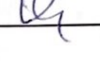
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи «5» червня 2024 р.

3. Вихідні дані до роботи: технічне та програмне забезпечення ЕОМ, вимоги до розробки програмного забезпечення, ергономічні вимоги до функціонування програмного засобу.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити):
Аналіз сучасного стану проблеми, існуючих методів і засобів її розв'язання, аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення, опис функціонального наповнення об'єкта проектування, практична реалізація об'єкта проектування.

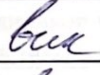
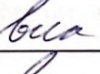
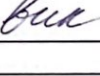
5. Перелік графічного матеріалу: 27 рисунків.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Андрущак І. Є.		
Специфікації вимог до розробленої системи	Андрущак І. Є.		
Розробка об'єкта проектування	Андрущак І. Є.		
Нормоконтроль	Повстяна Ю. С.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту		6,3%	
Академічна доброчесність	Яциук А. А.		

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

КАЛЕНДАРНИЙ ПЛАН

№ 3/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	07.02.2024 р.	
2	Аналіз проблеми розробки та впровадження об'єкту проектування	27.02.2024 р.	
3	Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання	12.03.2024 р.	
4	Розробка функціонально-структурної схеми роботи об'єкта проектування та проектування бази даних	17.04.2024 р.	
5	Практична реалізація об'єкта проектування та розробка бази даних	18.05.2024 р.	
6	Тестування та налагодження об'єкта проектування	01.06.2024 р.	
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедру	05.06.2024 р.	

Здобувач вищої освіти

Керівник кваліфікаційної роботи


(підпис)


(підпис)

Гузоватий С. В.
(прізвище, ініціали)

Андрущак І. Є.
(прізвище, ініціали)

**Рецензія
на кваліфікаційну роботу**

Здобувач вищої освіти: Гузоватий Сергій Володимирович

Тема: «Розробка сайту-візитки для компанії з міжнародних перевезень «MPL»»

Коротка характеристика кваліфікаційної роботи: Кваліфакаційна робота бакалавра складається з трьох розділів, в яких було проведено аналіз сучасного стану проблеми, чітко поставлено завдання, описана структура сайту та методи його реалізації, здійснено розробку і проведено функціональне тестування, а розгорнуто сайт на хостингу.

Самостійні розробки і пропозиції автора: Автором був самостійно розроблений сайт-візитка для логістичної компанії.

Практичне значення роботи Полягає в проектуванні та розробці сайту-візитки для логістичної компанії.

Недоліки: Суттєвих недоліків в роботі не виявлено.

Загальний висновок: Кваліфікаційна робота бакалавра виконана на високому рівні. Матеріал викладено чітко і зрозуміло. Вважаю, що кваліфакаційна робота відповідає усім вимогам і заслуговує оцінки «відмінно», а її авторові, студенту Гузоватому С.В., може бути присвоєний кваліфікаційний рівень “бакалавр” зі спеціальності «Інженерія програмного забезпечення».

Рецензент: к.т.н., доц. Кошечук В.А.

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**

Здобувач вищої освіти Гузоватий Сергій Володимирович
(прізвище, ім'я, по батькові)
група ІПЗ-41

Тема кваліфікаційної роботи «Розробка сайту-візитки для компанії з міжнародних перевезень «MPL»»

Керівник Андрущак Ігор Євгенович

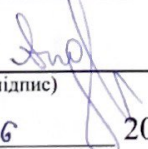
Актуальність теми Робота актуальна та своєчасна. Сайти, в тому числі і сайт-візитка, допомагає в просуванні компанії в інтернеті та представленні своїх послуг.

Об'єкт дослідження Набори методів та способів розробки сайтів

Характеристика теоретичного рівня, наявності самостійних розробок і практичної значимості роботи Автор аналізує різні методи та способи розробки сайтів. Визначає їх переваги та недоліки і обґрунтовує свої рішення. Чітко описує структуру сайту та методи для його реалізації. Зміст роботи повністю відповідає обраній темі.

Зауваження та недоліки: Суттєвих недоліків в роботі не виявлено.

Загальний висновок: Кваліфікаційна робота бакалавра виконана на високому рівні. Всі вимоги було дотримано, а робота описана логічно та послідовно. Пояснювальна записка відповідає усім стандартам оформлення.

Керівник 
(підпис)

(Андрущак І. Є.)
(прізвище, ім'я, по батькові)

«7» 06 2024 р.

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-ВІЗИТКИ ДЛЯ КОМПАНІЇ З
МІЖНАРОДНИХ ПЕРЕВЕЗЕНЬ «MPL»

DEVELOPMENT OF A BUSINESS CARD WEBSITE FOR THE
INTERNATIONAL TRANSPORTATION COMPANY "MPL"

спеціальність 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

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

Виконав: здобувач вищої освіти
Групи ІПЗ-41
Гузоватий Сергій Володимирович

(підпис)

Керівник:
д.т.н., професор
Андрущак Ігор Євгенович

(підпис)

Кваліфікаційну роботу
допущено до захисту
«__» _____ 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

«__» _____ 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Гузоватому Сергію Володимировичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Розробка сайту-візитки для компанії з міжнародних перевезень «MPL»

Керівник роботи: д.т.н., професор, Андрущак Ігор Євгенович

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 програмного забезпечення, ергономічні вимоги до функціонування програмного засобу.

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

Аналіз сучасного стану проблеми, існуючих методів і засобів її розв'язання, аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення, опис функціонального наповнення об'єкта проектування, практична реалізація об'єкта проектування.

5. Перелік графічного матеріалу: 27 рисунків.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Андрущак І. Є.		
Специфікації вимог до розробленої системи	Андрущак І. Є.		
Розробка об'єкта проєктування	Андрущак І. Є.		
Нормоконтроль	Повстяна Ю. С.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту		____%	
Академічна доброчесність	Яцук А. А.		

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

КАЛЕНДАРНИЙ ПЛАН

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

Здобувач вищої освіти

(підпис)

Гузоватий С. В.
(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис)

Андрущак І. Є.
(прізвище, ініціали)

АНОТАЦІЯ

Гузоватий С. В. Розробка сайту-візитки для компанії з міжнародних перевезень «MPL». Рукопис.

Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення» спеціальності 121 Інженерія програмного забезпечення. Луцький національний технічний університет. Луцьк, 2024.

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, висновків, списку використаних джерел (згідно структури кваліфікаційної роботи, затвердженої кафедрою).

У першому розділі здійснено аналіз сучасного стану проблеми розробки сайтів і сформульовано завдання. В другому розділі визначено вимоги до розроблюваної системи, а також засоби, методи і алгоритми розробки. У третьому розділі розроблено сайт-візитку для компанії з міжнародних перевезень «MPL». У висновках узагальнено інформацію, відображену у попередніх частинах.

Результати розробки демонструють можливості певних технологій у створенні веб-сайтів.

Ключові слова: веб-сайт, сайт-візитка, back-end, front-end, база даних, фреймворк, Laravel, React.js, Inertia.js.

ABSTRACT

Huzovaty S. V. Development of a Business Card Website for the International Transportation Company "MPL". Manuscript.

Bachelor's qualifying thesis of the educational program "Software Engineering". Lutsk National Technical University. Lutsk, 2024.

The bachelor's qualification work consists of an introduction, three sections, conclusions, a list of used sources (according to the structure of the qualification work approved by the department).

In the first chapter, an analysis of the current state of the problem of developing websites was carried out and the task was formulated. The second section defines the requirements for the developed system, as well as means, methods and development algorithms. In the third chapter, Business Card Website for the International Transportation Company "MPL" was developed. The conclusions summarize the information presented in the previous parts.

The development results demonstrate the possibilities of specific technologies in creating websites.

Keywords: website, business card, back-end, front-end, database, framework, Laravel, React.js, Inertia.js.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ СУЧАСНОГО СТАНУ ПРОБЛЕМИ І ПОСТАВНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ	9
1.1 Аналіз сучасного стану проблеми	9
1.2 Постановка завдання на кваліфікаційну роботу бакалавра	11
Висновки до розділу 1	12
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ	13
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проєктування програмного забезпечення	13
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання	15
Висновки до розділу 2	21
РОЗДІЛ 3 РОЗРОБКА САЙТУ-ВІЗИТКИ ДЛЯ КОМПАНІЇ З МІЖНАРОДНИХ ПЕРЕВЕЗЕНЬ	23
3.1 Практична реалізація об'єкта проєктування	23
3.2 Тестування та налагодження інформаційно-комп'ютерної системи ...	34
3.3 Розробка бази даних.....	37
3.4 Розгортання сайту на хостингу	39
3.5 Захист інформаційно-комп'ютерної системи.....	41
Висновки до розділу 3	43
ВИСНОВКИ.....	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	46

ВСТУП

Актуальність теми. Кожного дня, інтернет охоплює все більше сфер діяльності людини, тому звичайні методи поширення інформації можуть бути не достатньо ефективними, особливо це стосується розвитку якоїсь власної справи. Сьогодні, майже кожен бізнес володіє своїм веб-сайтом. Так склалося, що зараз, наявність сайту приносить власникам компаній велику перевагу, у вигляді швидшого розвитку чи збільшенні доходів.

Якщо говорити про доцільність розробки сайту-візитки логістичної компанії, то у час розвитку інтернету, це стає необхідністю. Варто відзначити невеликі або новостворені компанії, для яких наявність навіть простого та не сильно функціонального сайту дає суттєву перевагу – це чудовий спосіб самореклами для залучення нових клієнтів, які зможуть дізнатись про компанію та її послуги. Також сайт-візитка може використовуватись як міні-платформа для спілкування з клієнтами, наявність якої, дозволить відвідувачам швидко зв'язатися з компанією.

Метою роботи є аналіз та визначення головних методів та способів розробки сайтів, веб-додатків.

Завдання дослідження сформовано та виглядає наступним чином:

- визначити актуальність проблеми та потребу в створенні сайту-візитки для логістичної компанії;
- визначити функціональні вимоги до розроблюваного сайту-візитки;
- розробити функціональну структуру майбутнього інтернет-ресурсу та схему його сторінок, описати їх вміст;
- провести аналіз способів та методи їх реалізації в напрямку розробки веб-сайтів;
- зробити обґрунтований вибір існуючих способів та технологій, які будуть використані для подальшої розробки, провести їх додатковий аналіз;
- реалізувати сайт-візитку для компанії з міжнародних перевезень;
- провести функціональне тестування під час розробки сайту;

– розгорнути розроблюваний сайт на хостингу.

Об'єктом кваліфікаційної роботи бакалавра є набір методів та способів розробки сайтів.

Предметом кваліфікаційної роботи бакалавра є сайт-візитка для компанії з міжнародних перевезень «MPL».

РОЗДІЛ 1

АНАЛІЗ СУЧАСНОГО СТАНУ ПРОБЛЕМИ І ПОСТАВНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

Ми живемо в світі, де логістика є ключовим елементом сучасної економіки та бізнесу. Ефективне управління логістичними операціями забезпечує постійне функціонування різних галузей промисловості та торгівлі. Загалом, від компаній, які займаються різними перевезеннями, залежить підтримка стабільності або ж розвитку економічної складової регіону чи країни [1].

Існує багато видів логістики, так склалося через те, що їх виділять за напрямом реалізації, як-от комерційна, розподілу, поставок тощо. Для повної розкриття теми, потрібно розібратися саме в транспортній логістиці.

Транспортна логістика є основним елементом в логістичних процесах, вона зосереджується на плануванні та доставці вантажу з одного місця в інше, при цьому може використовуватись будь-який вид транспорту. Її ж метою є забезпечення клієнта необхідною кількістю транспортних засобів для перевезення оголошеного вантажу, в зазначене місце та час [2].

Способами реалізації транспортної логістики є автомобільний, залізничний, морський та повітряний транспорт, на вибір способу перевезення впливають габарити та вага вантажу, вартість доставки, наявність необхідної інфраструктури.

У сфері логістики, в Україні до 24 лютого 2022 року, найважливішими способами перевезень були морський та залізничний транспорт, проте з початком повномасштабної війни, велику частку ринку транспортування зайняв саме автомобільний транспорт [3]. Для багатьох компаній, що займають міжнародними чи внутрішніми перевезеннями, основною реалізацією своїх послуг стала саме автомобільна логістика. Перевагами автомобільних вантажоперевезень є швидкість, доступність, зручність та головне те, що

використовується принцип доставки від пункту завантаження до точки призначення [4].

Для сталого розвитку компанії, яка займається перевезеннями, та підвищення її конкурентоспроможності, почали впроваджувати певні логістичні стратегії. Вони повинні визначати певні аспекти, які в подальшому будуть покращуватись [5]. Для невеликих або новостворених компаній, доцільним буде виконання таких завдань:

- зростання обсягів використання послуг;
- вихід на нові ринки;
- поліпшення комунікації між компанією та споживачами послуг.

З такою логістичною стратегією, допоможе створення веб-сайту для компанії. Сайт та інформація на ньому, буде демонструвати свій власний бренд та рекламувати послуги перевезення.

Існує багато видів сайтів, але як основні, виділяють:

- корпоративний сайт;
- інтернет-магазин;
- landing page;
- сайт-візитка.

Для невеликих або новостворених логістичних компаній, варто почати із сайту-візитки, це буде чудовим стартом для подальшого розвитку, адже при створенні такого сайту є свої переваги:

- простота, зазвичай складається з кількох сторінок та має просту структуру, при цьому зменшується час розробки, в порівнянні з іншими видами сайтів. Це дозволяє компанії швидко продемонструвати себе в мережі інтернет;
- економічність, розробка та підтримка такого сайту є менш ресурсозатратною, для невеликих та новостворених компаній, це важливий показник;
- нові ринки та клієнти, введений в експлуатацію сайт, буде містити основну інформацію про компанію, послуги та контакти. Це дозволить легше отримувати нових клієнтів та підтримувати з ними комунікацію.

Розробка сайту-візитки для логістичної компанії, особливо новостворена, є важливим кроком для подальшого розвитку, адже дає певні переваги над тими, у кого його немає. У тому випадку, якщо компанія вже має сайт, розробка нового є недоцільним використанням часу та коштів.

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

Під час аналізу предметної області, було визначено, що логістичні компанії відіграють важливу роль як на регіональному, так і на міжнародному рівні, тому для невеликих або новостворених компаній є необхідністю створення представлення себе в мережі Інтернет, адже це дозволяє швидко залучити нових клієнтів та створити умови для успішного ведення бізнесу.

Результатом цієї кваліфікаційної роботи бакалавра має стати розроблений сайт-візитка, який буде представляти логістичну компанію з міжнародних перевезень в інтернеті. Сайт повинен містити повну інформацію про компанію, рішення для спілкування з відвідувачами ресурсу та можливістю змінювати дані від автентифікованого користувача, який має певні права доступу.

Посилаючись на те, що має містити сайт, в ході виконання кваліфікаційної роботи бакалавра, поставлено наступні задачі:

- визначити актуальність проблеми та потребу в створенні сайту-візитки для логістичної компанії;
- визначити функціональні вимоги до розроблюваного сайту-візитки;
- розробити функціональну структуру майбутнього інтернет-ресурсу та схему його сторінок, описати їх вміст;
- провести аналіз способів та методи їх реалізації в напрямку розробки веб-сайтів;
- зробити обґрунтований вибір існуючих способів та технологій, які будуть використані для подальшої розробки, провести їх додатковий аналіз;
- реалізувати сайт-візитку для компанії з міжнародних перевезень;
- провести функціональне тестування під час розробки сайту;

- розгорнути розроблюваний сайт на хостингу.

Висновки до розділу 1

В даному розділі, був проведений аналіз предметної області, результат якого дозволив зрозуміти важливість логістичних операцій в економіці, забезпечуючи її стабільність або розвиток. Було досліджено важливість розробки сайту-візитки, особливо для невеликих або новостворених компаній, створення сайту – це основа для успішного ведення справи, перевагами якого є простота та економічність.

Був сформований та поставлений майбутній результат цієї кваліфікаційної роботи бакалавра у вигляді вмісту та функціоналу розроблюваного сайту. На основі цього, написані задачі, які повинні бути досягнені по завершенню виконання роботи.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

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

Функціональні вимоги – це важлива частина в розробці програмного продукту, вони описують як саме повинна функціонувати система, які функції мусить виконувати, права доступу для різних груп користувачів тощо.

Зазвичай перед розробкою певного програмного продукту будують модель, яка візуалізує функціональні вимоги до розроблюваної системи. Для цього можуть використовувати діаграму варіантів використання, інша назва – діаграма прецедентів [6].

Для візуалізації вимог розроблюваного сайту-візитки логістичної компанії, було створено діаграму прецедентів (рис. 2.1).

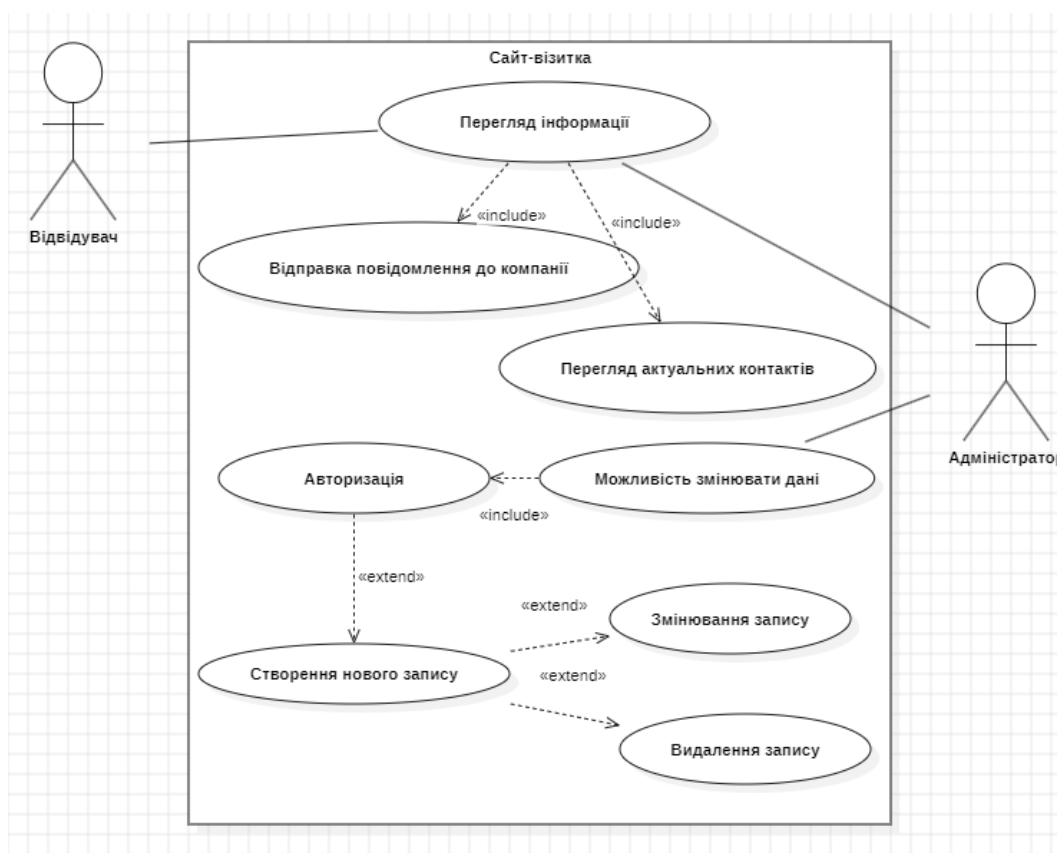


Рисунок 2.1 – Діаграма прецедентів

Діаграма прецедентів має два актора:

- відвідувач – це особа, яка може переглядати інформацію про компанію, включаючи актуальні контакти та користування формою зв'язку для відправки повідомлень на пошту;
- адміністратор – це особа, яка має можливість змінювати дані на сайті, авторизувавшись на ньому, також доступні сценарії відвідувача.

Розроблюваний сайт-візитка повинен мати кілька сторінок, на яких буде міститися повна інформація про компанію. Структура майбутнього інтернет-ресурсу складатиметься з п'яти сторінок (рис. 2.2).

Одна з сторінок – це «Головна», де буде короткий вступ та описи з посиланнями на інші елементи, такі як:

- «Послуги» – опис послуг, які надає компанія, та додаткові переваги при співпраці;
- «Про нас» – інформація про компанію, її цінності та короткий текст про партнерів;
- «Наш транспорт» – інформаційні картки з транспортом компанії;
- «Контакти» – адреса, номер та електронна пошта компанії, форма для зворотнього зв'язку.

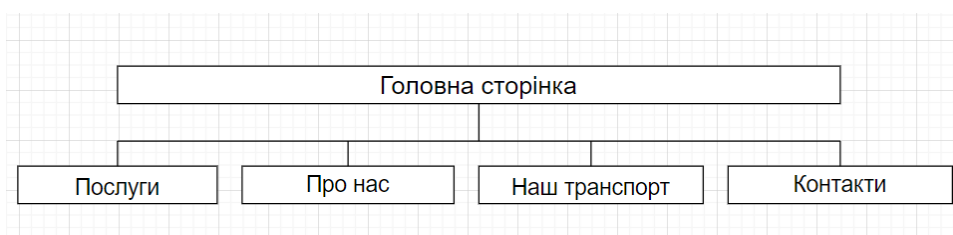


Рисунок 2.2 – Структура сайту-візитки

Кожна сторінка сайту буде розроблятися за схемою, яка зображена на рисунку 2.3, де:

- «header», містить логотип компанії, навігацію по сайту та кнопку авторизації для адміністратора;
- «main», містить вміст сторінки;

- «footer», містить повідомлення про авторське право на сайт.

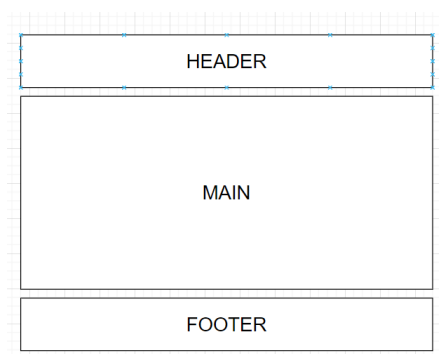


Рисунок 2.3 – Схема сторінок

Отже, було описано функціональні вимоги, структуру сайту та схему майбутніх сторінок, вміст яких повністю висвітлює актуальну інформацію про логістичну компанію.

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Перед початком розробки, потрібно визначитись із основними факторами, які впливають на вибір метода створення сайту – це може бути об’єм та вид інтернет-ресурсу, кількість функціоналу, який потрібно додати, час, виділений на розробку, або витримування великих навантажень тощо.

Метод розробки сайтів вручну або ж розробка з нуля. Цей спосіб передбачає створення веб-сайту, використовуючи HTML та CSS, за необхідності JavaScript.

HTML – це мова розмітки, яка використовується для написання структури та вмісту веб-сторінки. Написаний код зберігається як HTML-файл та надалі може бути опрацьований браузером для перегляду готової сторінки [7].

CSS – мова стилів, створена для того, щоб розділити зовнішній вигляд веб-сторінки від наповнення у файлі, написаного на мові розмітки. З його

використанням під час розробки, HTML відповідає тільки за структуру, а CSS за візуальне представлення сторінки [8].

JavaScript – на відміну від HTML та CSS, є мовою програмування. Він використовується для написання певних сценаріїв поведінки сторінки: анімації, оновлення даних, створення інтерактивних елементів, тобто добавляє веб-сторінці динамічності [9].

В загальному, HTML, CSS, JavaScript є трьома головними елементами при розробці веб-сайтів. Якщо повернутися до методу розробки, то використання цих трьох мов забезпечує створення чудового статичного сайту, за потреби, з певними елементами динамічності. Але для веб-ресурсу з великою кількістю сторінок, краще вибрати інший та простіший спосіб його втілення.

Наступним методом розробки сайтів є використання CMS. CMS (Content Management System) – система керування вмістом, яка надає користувачеві інтерфейс, що набагато спрощує створення сайту.

CMS складається з трьох елементів [10]:

- інтерфейс, з яким взаємодіє користувач для створення сторінок чи їх редагування або ж налаштування самої CMS;
- база даних, тут зберігаються весь вміст сторінок, та різні налаштування;
- веб-сервер, може знаходитись на хостингу або ж запущеним локально, тут розміщена CMS, база даних та відбувається обробка різних запитів.

Переваги використання CMS над іншими методами розробки:

- можливість створення сайтів людям, які не володіють високим рівнем знань в програмуванні;
- менша вартість розробки;
- менше витраченого часу на розробку сайтів.

У підсумку, використання CMS є чудовим рішенням для простого та швидкого створення великого спектру сайтів.

Останнім методом розробки сайтів є використання фреймворків. Це допоміжний інструмент розробки, який надає основу майбутньому проєкту.

Містить в собі набори бібліотек та інструментів, які спрощують розробку програмного забезпечення [11].

У сфері веб-розробки, фреймворки можна поділити на дві групи:

- backend, працюють з серверною частиною сайту;
- front-end, працюють з клієнтською частиною сайту.

Back-end розробка включає в себе створення логіки та обробки даних зі сторони серверної частини сайту та керування базою даних. Переваги використання back-end фреймворків:

- швидка розробка серверної частини, фреймворки надають основу і пришвидшують подальшу розробку;
- безпека, фреймворки надають уже перевірені та надійні функції та методи безпеки, також присутні регулярні їх оновлення;
- інтегрованість, використання готових рішень як RESTful Api, OAuth2 тощо.

Недоліки використання back-end фреймворків:

- висока складність для людей, які тільки починають працювати з фреймворками і не мають базових знань в програмуванні;
- можливий надмірний розмір проєкту, який створено на основі фреймворку.

Back-end фреймворки, які популярні станом на січень 2023 року:

- Laravel;
- Django;
- Ruby on Rails.

Laravel – безкоштовний фреймворк, який використовує мову програмування PHP та архітектуру MVC (Model-View-Controller). Вважається найпопулярнішим упродовж кількох останніх років.

Популярність цього фреймворка наряду пов'язана з тим, що більшість сайтів, близько 70 %, написані мовою програмування PHP [12]. Переваги, які виділяють при використанні Laravel:

- наявність потужного інструментарію для роботи з базою даних;

- вбудовані інструменти для реалізації авторизації користувачів та функцій безпеки;

- використання найновіших версій PHP;
- наявність великої та добре написаної документації.

Недоліки, які виділяють при використанні Laravel:

- постійні оновлення можуть створити труднощі для підтримки проєкту;
- легкий у вивченні, завдяки документації, але в практичній реалізації тяжкий, особливо для початківців.

Django – популярний фреймворк, який використовує мову програмування Python та архітектуру MVT (Model-View-Template).

Переваги, які виділяють при використанні Django [13]:

- універсальність, може використовуватись для розробки майже будь-яких сайтів чи додатків;

- наявність різних інструментів та функцій безпеки;
- проєкти Django будуються таким чином, що мають високий рівень масштабованості;

- наявність великої документації.

Недоліки, які виділяють при використанні Django:

- порівняно з іншими фреймворками, має застарілу систему, яка працює з базами даних;
- має свою специфічну файлову структуру;
- не найкраще рішення для невеликих проєктів.

Ruby on Rails – популярний фреймворк, який використовує мову програмування Ruby та архітектуру MVC.

Переваги, які виділяють при використанні Ruby on Rails:

- великий вибір інструментів та бібліотек;
- встановлюється з багатьма плагінами та модулями, що значно зменшують час розробки.

Недоліки, які виділяють при використанні Ruby on Rails:

- має меншу продуктивність ніж інші фреймворки;

- відсутність гнучкості, налаштування різних компонентів та модулів відбувається при встановленні фрейворка у проєкт, недоліки, які можуть з'явитися під кінець розробки, важко виправити;
- вивчення документації може викликати труднощі.

Кожен з фреймворків, які були розглянуті вище, мають відкритий вихідний код та всі використовують подібну архітектуру, їх було доцільно порівнювати між собою. Загалом, вони використовуються для розробки великого спектру сайтів та набагато спрощують роботу над його серверною частиною. Було розглянуто всього лиш кілька популярних фреймворків, хоча їх є багато (рис. 2.4).

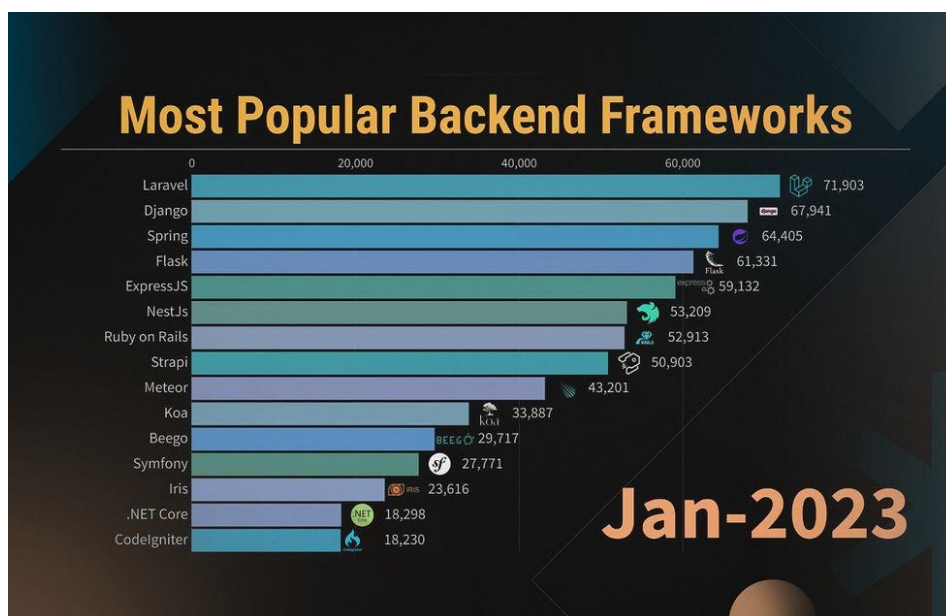


Рисунок 2.4 – Рейтинг популярності back-end фреймворків [14]

Front-end розробка включає в себе створення веб-інтерфейсу для користувача, а з використанням фреймворків, його можна зробити динамічним та достатньо оптимізованим під велику кількість даних.

Між front-end фреймворками, варто виділити React.js та Vue.js. Вони є частиною так названої «JavaScript екосистеми» та використовують компонентний підхід у розробці. Завдяки компонентам, зменшується час на розробку сайтів, а також кількість написаного коду.

Проаналізувавши способи розробки сайтів та їх інструменти, потрібно визначитись з тими технологіями, які будуть використовуватись для виконання кваліфікаційної роботи бакалавра.

Метод розробки, використання фреймворків, це не найлегший спосіб, але якщо під час роботи над проєктом, або ж в майбутньому, потрібно буде багато унікального функціоналу, то будь-який фреймворк впорається краще ніж CMS.

Серверна частина сайту, це його основа, тому проаналізувавши переваги та недоліки різних back-end фреймворків, було обрано Laravel.

Є два способи як можна використовувати Laravel:

- використання як full stack фреймворк, що передбачає його використання у двох напрямках розробки, як серверної, так і клієнтської. Також з використанням сторонніх пакетів, можна підключити додаткові front-end рішення, а не користуватися тільки вбудованими інструментами Laravel;

- використання як API back-end, передбачає написання серверного інтерфейсу для програм, які створені front-end фреймворками. В даному випадку, Laravel буде відповідати тільки за обробку та взаємодію з даними [15].

Laravel надає розробникам великий набір інструментів та функцій, які спрощують процес створення сайтів, як основні виділяють:

- eloquent ORM – інструмент для взаємодії з базою даних, з його використанням, кожна таблиця має свою проекцію у вигляді моделі, яка дозволяє отримувати, добавляти, оновлювати чи видаляти дані;

- система міграцій – використовується як контроль версій бази даних, дозволяє просто добавляти нові таблиці чи поля або ж виконати відкат останніх змін, які зробив розробник;

- вбудовані інструменти для авторизації – значно спрощує розробку системи аутентифікації користувачів, дозволів та захист сайту. При цьому, розроблений інтернет-ресурс буде повністю захищеним завдяки перевіреним рішенням, які надає Laravel;

- шаблонізатор Blade – це інструмент для створення представлень сторінок, який надає Laravel. Це зручний спосіб для роботи з шаблонами, тому

що надається використання PHP коду на сторінках сайту та певні функції для HTML.

Для розробки сайту, було вирішено не обмежуватись тільки інструментами, які надає Laravel, тому, як front-end фреймворк, обрано React. Він дуже подібний до Vue, але одне з останніх оновлень, зробило його значно складнішим, тому цей варіант відсіявся сам по собі.

Із переваг використання React виділяють:

- висока продуктивність, вона досягається за рахунок використання VDOM, яка зберігає інтерфейс користувача в оперативній пам'яті;
- компонентна структура, дозволяє створювати певні елементи, які можна повторно використовувати, це пришвидшує та спрощує процес розробки сайту;
- наявність багатьох інструментів, які спрощують налаштування та оптимізацію сайту;
- має простий синтаксис та чітку документацію.

Якщо обговорювати недоліки, то виділяють: складність інтеграції в існуючі сайти та необхідний високий рівень знань для роботи з React.

Виходить так, що для створення сайту, будуть використовуватись два різних фреймворка. Для їх нормального функціонування між собою, необхідно підключити Inertia.js. Якщо коротко, то це бібліотека, яка дозволяє розробляти сайти без написання API, вона буде зв'язувати React.js та Laravel.

Висновки до розділу 2

В даному розділі, була побудована UML діаграма, яка описує функціональні вимоги до розроблюваного сайту-візитки. Також описана та візуалізована структура сайту і схема сторінки, описано вміст кожного елементу майбутнього інтернет-ресурсу.

Було коротко розглянуто способи розробки веб-сайтів, проаналізовано back-end фреймворки та описані їх переваги і недоліки. Був зроблений вибір

технологій, які будуть задіяні в розробці, він ґрунтується на основі аналізу переваг та недоліків різних фреймворків.

Був зроблений додатковий збір інформації про технології, які було вибрані. Laravel – було наведено способи використання та інструменти, які надаються розробникам. React – виділено переваги та недоліки цього front-end фреймворку, так під час аналіз був тільки короткий опис. Використання вибраних технологій дозволяє побудувати високоякісний сайт та дає можливість для масштабування.

РОЗДІЛ 3

РОЗРОБКА САЙТУ-ВІЗИТКИ ДЛЯ КОМПАНІЇ З МІЖНАРОДНИХ ПЕРЕВЕЗЕНЬ «MPL»

3.1 Практична реалізація об'єкта проєктування

Першим кроком розробки сайту є встановлення та налаштування фреймворків. Так як основа проєкту це Laravel, то процес створення починається з нього. Для швидкого старту, необхідно використовувати вбудований інструмент Sail, який допомагає працювати з технологією Docker та відразу налаштує його конфігураційні файли. Для самого встановлення, була використана команда, яка наведена в лістингу 3.1.

Лістинг 3.1 – Команда встановлення проєкту

```
curl -s https://laravel.build/MPL | bash
```

Кінець лістингу 3.1

Після встановлення Laravel, потрібно інсталиювати деякі пакети, які будуть використовуватись для зв'язування front-end та back-end фреймворків. Початок цієї частини буде з інструменту Vite, він буде забезпечувати збір та обробку файлів React, налаштований файл конфігурації наведено в лістингу 3.2.

Лістинг 3.2 – Налаштований файл «vite.config.js»

```
import {defineConfig} from 'vite';
import laravel from 'laravel-vite-plugin';
import react from '@virejs/plugin-react';
export default defineConfig({
  plugins: [
    react(), //react функції та дозвіл до jsx файлів.
    laravel({
      input:['resources/css/app.css', 'resources/js/app.jsx'],
      refresh: true, //автоматичне оновлення браузера при змінах
    }),
  ],
});
```

Кінець лістингу 3.2

Наступним кроком є налаштування Inertia.js, вона забезпечує візуалізацію сторінок React. Для серверної частини – це визначення основного HTML документа, в якому і будуть відкриватися розроблювані сторінки, код налаштованого файлу наведено в лістингу 3.3.

Лістинг 3.3 – Файл «app.blade.php»

```
<html>
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0" />
    @viteReactRefresh
    @vite(['resources/css/app.css', 'resources/js/app.jsx'])
    @inertiaHead
  </head>
  <body>
    @inertia //місце візуалізації сторінок
  </body>
</html>
```

Кінець лістингу 3.3

Налаштування Inertia.js для клієнтської частини – це ініціалізація кореневого файлу для front-end фреймворку, його завдання – завантажувати створені в майбутньому сторінки React. Результат налагодження наведено в лістингу 3.4.

Лістинг 3.4 – Файл «app.jsx»

```
import React from 'react'
import { createInertiaApp } from '@inertiajs/react'
import { createRoot } from 'react-dom/client'
createInertiaApp({
  resolve: name => { { /*завантаження сторінок*/ }
    const pages = import.meta.glob('./Pages/**/*.jsx', { eager: true })
    return pages[`./Pages/${name}.jsx`]
  },
  setup({ el, App, props }) { { /*налаштування та рендер додатку*/ }
    createRoot(el).render(<App {...props} />)
  },
})
```

Кінець лістингу 3.4

З усіма налаштуваннями, вибудована наступна логіка:

- laravel створює одну HTML сторінку, де підключається Vite та Inertia.js;
- vite компілює файли React;
- inertia.js візуалізує скомпільовані файли у сторінці Laravel.

Згідно з структурою сторінки, де є три елемента – це «header», «main» та «footer», доцільно створити певний шаблон, який буде використовуватись до всіх сторінок. Він буде створений як компонент React, де змінним контентом є тільки вміст сторінки, тобто навігація та авторське право – статичним.

Функція «GuestLayout», яка наведена в лістингу 3.5, буде використовуватись на всіх сторінках, має приймати одне значення, а саме «children» – це контент майбутньої сторінки. Всі теги у компоненті стилізовані таким чином, що «header» завжди прикріплений зверху, а «footer» – знизу екрана.

Лістинг 3.5 – Демонстрація функції «GuestLayout»

```
export default function GuestLayout({children}) {
  return (
    <div className="flex flex-col min-h-screen">
      <header className="w-full shadow-lg sticky top-0 z-50">
        /*вміст «шапки» сторінки */
      </header>
      <main className="mb-auto">{children}</main>
      <footer className="mt-auto bottom-0 w-full">
        /*вміст «підвалу» сторінки*/
      </footer>
    </div>
  );
}
```

Кінець лістингу 3.5

У тег «header», добавлено кнопки навігації та деяка інформація, а саме адреса та номер телефону (рис. 3.1). В елемент «footer», був добавлений текст про авторське право.

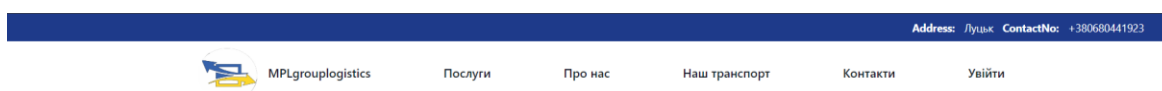


Рисунок 3.1 – Елемент «header»

Після завершення створення шаблону, можна розробляти інші сторінки та заповнювати їх інформацією. Першою буде головна сторінка, яка має містити короткий опис про компанію.

Головна сторінка умовно поділена на чотири блоки. Кожен має містити коротку інформацію про інші елементи структури сайту та посилання на них.

Перший блок – містить основну фотографію та короткий опис компанії (рис. 3.2).

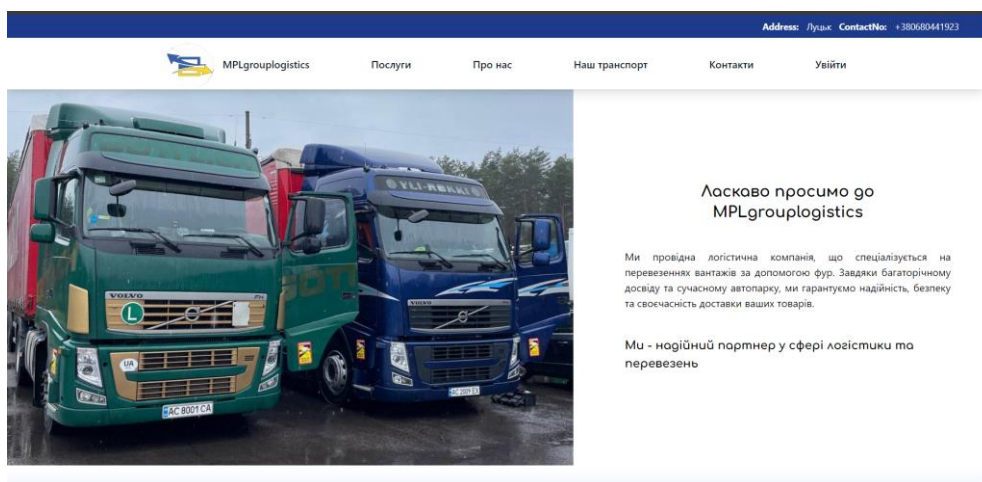


Рисунок 3.2 – Перший блок головної сторінки

Сторінка «Про нас» – складається тільки з текстової інформації: історія, цінності, партнери (рис. 3.3).

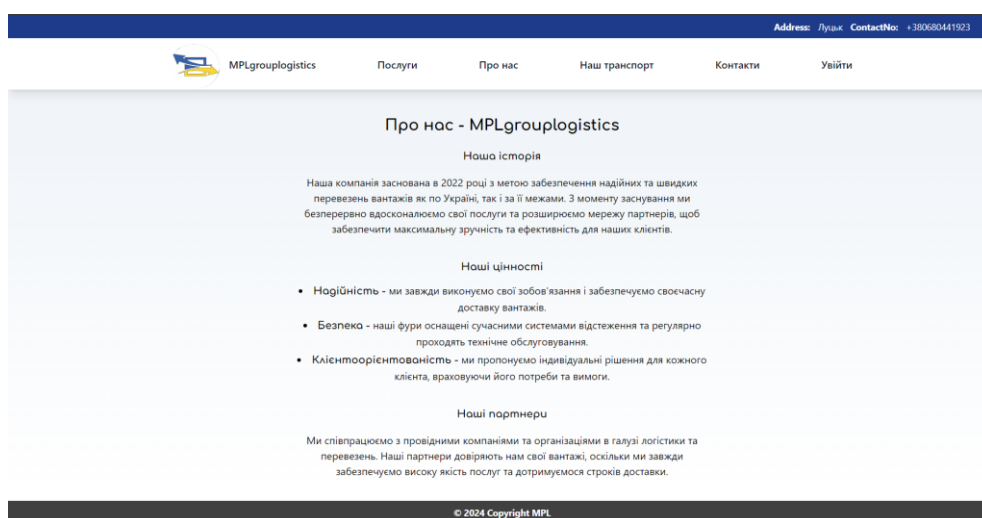


Рисунок 3.3 – Сторінка «Про нас»

Маршрутизація для статичних сторінок, таких як «Про нас», налаштовується напряму, без створення контролерів та маніпулюванням з даними, код такого типу маршрутів наведено в лістингу 3.6.

Лістинг 3.6 – Демонстрація маршрутизації для статичної сторінки

```
Route::get('/about', function () {
    return inertia('About'); //рендер сторінки «Про нас»
});
```

Кінець лістингу 3.6

Другий блок – містить назви послуг та посилання на сторінку «Послуги» (рис. 3.4).

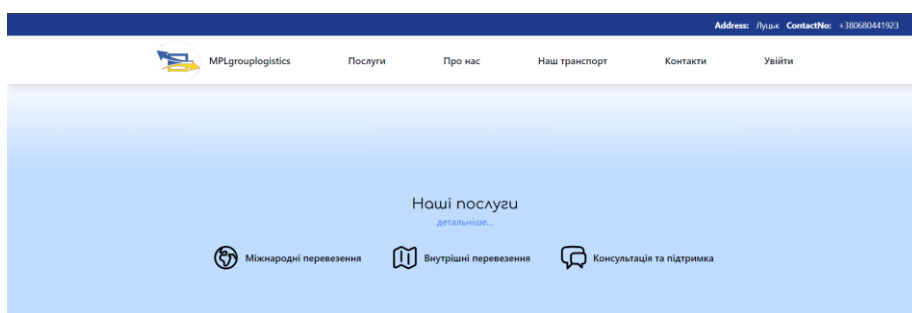


Рисунок 3.4 – Другий блок головної сторінки

Сторінка «Послуги» – містить опис послуг, які надає компанія, та додаткові переваги при співпраці (рис. 3.5).



Рисунок 3.5 – Сторінка «Послуги»

Перехід між блоками головної сторінки та іншими елементами структури інтернет-ресурсу, створений завдяки компоненту – «Link», який надає бібліотка Inertia.js. Він працює таким чином, що, при натисканні, не дозволяє повністю перезавантажити сайт та дозволяє працювати йому як SPA (Single Page Application). Використання компоненту «Link» наведено в лістингу 3.7.

Лістинг 3.7 – Демонстрація використання компонента «Link»

```
<Link href="/services"> {/*перехід до сторінки «Послуги»*/}
  <h6 className="mb-4 text-blue-500">детальніше...</h6>
</Link>
```

Кінець лістингу 3.7

Третій блок – містить короткий вступ до сторінки «Наш транспорт», разом з посиланням (рис. 3.6). Ключовою особливістю цієї частини, є написаний компонент, який перемикає фото транспорту – «CustomSlider».

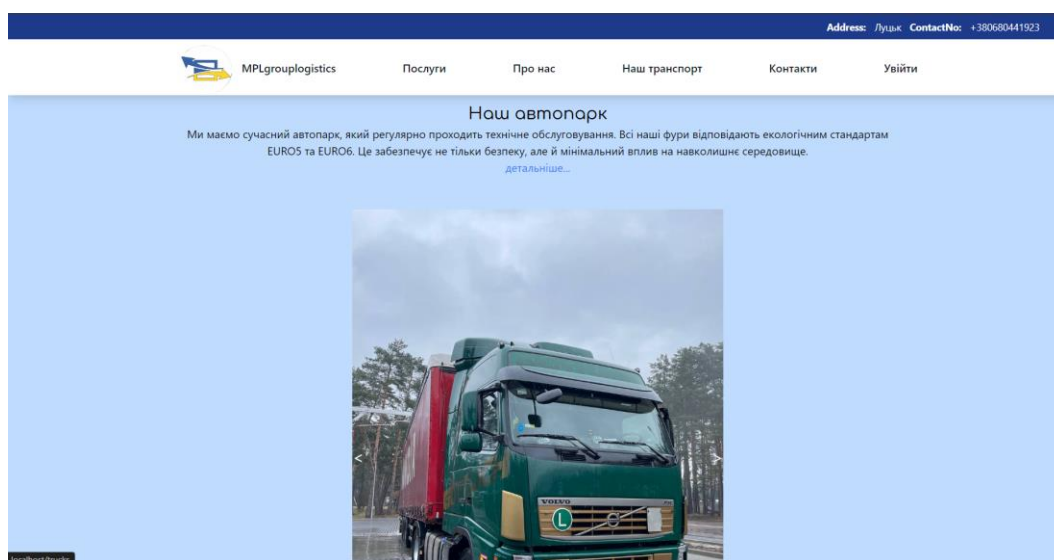


Рисунок 3.6 – Третій блок головної сторінки

Для реалізації компоненту «CustomSlider», був створений контролер, який тепер відповідає за маршрутизацію та передачу фотографій транспорту з бази даних. Код функції відправки значень наведено в лістингу 3.8.

Лістинг 3.8 – Демонстрація функції передачі даних до головної сторінки

```

class TrucksController extends Controller
{
    public function showImage()
    {
        return inertia('Main', [      //рендер головної сторінки
            'images' => Truck::pluck('image'), //Передача фотографій
        ]);
    }
}

```

Кінець лістингу 3.8

Тим часом, функція головної сторінки, буде приймати значення фотографій з бази даних та виводити їх через цикл в компоненті «CustomSlider». Використання «CustomSlider» наведено в лістингу 3.9.

Лістинг 3.9 – Демонстрація використання «CustomSlider»

```

<CustomSlider>
    {images.map((image, index) => {
        return <img key={index} src={"storage/" + image} alt={image} />;
    })}
</CustomSlider>

```

Кінець лістингу 3.9

Сторінка «Наш транспорт» – містить картки з певними характеристиками та фотографією транспорту (рис. 3.7).

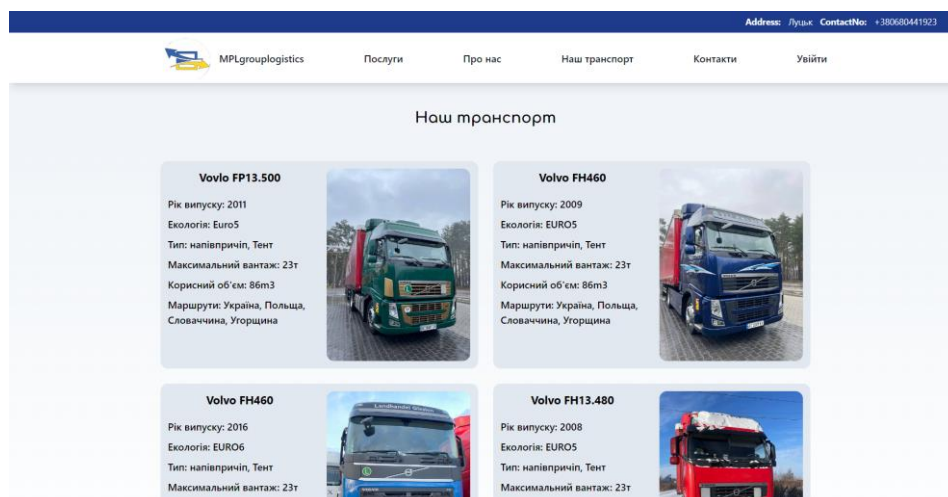


Рисунок 3.7 – Сторінка «Наш транспорт»

Так як дані, для сторінки «Наш транспорт», мають діставатися з бази даних, то була створена функція, яка буде передавати дані, її демонстрація наведена в лістингу 3.10.

Лістинг 3.10 – Демонстрація функції передачі даних до «Наш транспорт»

```
class TrucksController extends Controller
{
    public function show()
    {
        return inertia('Trucks', [
            'trucks' => Truck::all(), //Передача всіх даних
                                   //транспорту
        ]);
    }
}
```

Кінець лістингу 3.10

Четвертий блок – має кнопку для переходу на сторінку «Контакти», яка в свою чергу містить адресу, номер, електронну пошту, посилання на соцмережі компанії, а також форму зворотнього зв'язку (рис. 3.8, 3.9).

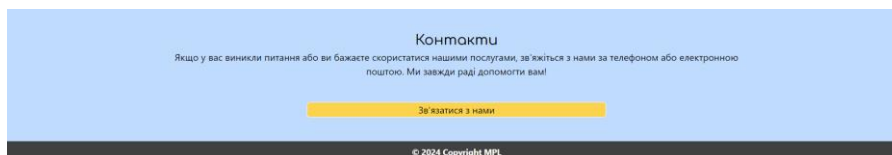


Рисунок 3.8 – Четвертий блок основної сторінки

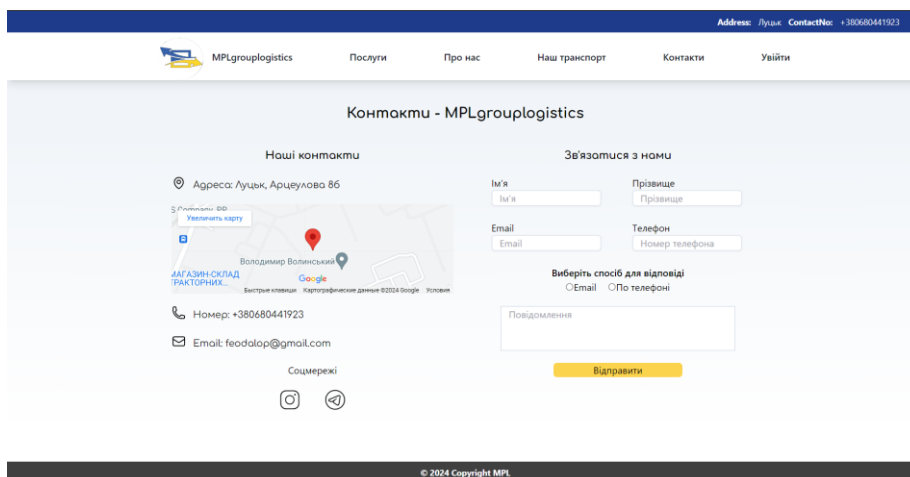


Рисунок 3.9 – Сторінка «Контакти»

Для реалізації відправки даних на пошту компанії, був створений контролер, де відбувається візуалізація відповідної сторінки та відправка повідомлення. Код функції, яка відправляє листи, наведено в лістингу 3.11.

Лістинг 3.11 – Демонстрація функції відправки повідомлення

```
public function mail(Request $request)
{
    $data = $request->validate([ //валідація даних з форми
        'firstName' => 'string',
        'lastName' => 'string',
        'email' => 'string',
        'phone' => 'string',
        'type' => 'string',
        'message' => 'string',
    ]);
    Mail::to(USER::EMAIL)->send(new ContactMail($data));
    return inertia('Contacts'); //рендер сторінки «Контакти»
}
```

Кінець лістингу 3.11

Дизайн повідомлення оформлюється як файл Blade, його код наведено в лістингу 3.12. В конфігураційному файлі проєкту налаштовано протокол пересилання пошти та її безпеку.

Лістинг 3.12 – Демонстрація файлу «mail.blade.php»

```
<!DOCTYPE html>
<html>
<head>
    <title>MPL - Форма зв'язку</title>
</head>
<body>
    <h1>Форма зв'язку</h1>
    <p>Ім'я та прізвище: {{ $data['firstName'] }}
    {{ $data['lastName'] }}</p>
    <p>Email: {{ $data['email'] }}</p>
    <p>Телефон: {{ $data['phone'] }}</p>
    <p>Зв'язатися: {{ $data['type'] }}</p>
    <p>Повідомлення: {{ $data['message'] }}</p>
</body>
</html>
```

Кінець лістингу 3.12

Наступним кроком була розробка функціоналу адміністратора. Був створений контролер, який відповідає за авторизацію користувача. Він приймає дані з форми авторизації та, завдяки вбудованим функціям Laravel, намагається увійти в систему. Код функції авторизації наведено в лістингу 3.13.

Лістинг 3.13 – Демонстрація функції авторизації

```
public function login(Request $request)
{
    $data = $request->validate([ //валідація даних
        'email' => 'required',
        'password' => 'required'
    ]);
    if (Auth::attempt($data)) { //спроба входу
        $request->session()->regenerate(); //створення сесії
        return to_route('main');
    }
    return to_route('main');
}
```

Кінець лістингу 3.13

Форма авторизації була реалізована через модальне вікно (рис. 3.10).

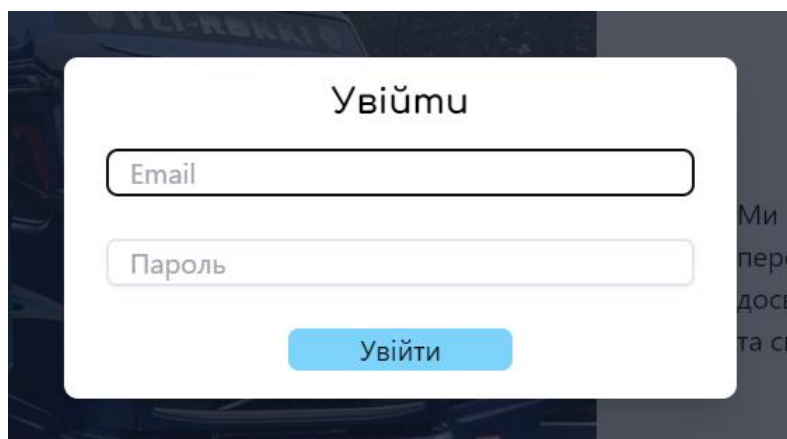


Рисунок 3.10 – Модальне вікно авторизації

Першим кроком для інтеграції функціоналу адміністратора на сторінках сайту є налаштування глобальних даних, які можуть передаватися до будь-якого компоненту. Це дозволяє зробити бібліотека Inertia.js зі своєю функцією «share».

Функція «share» була налаштована таким чином, що на кожному сторінку може приходити значення «auth», яке має містити роль користувача. Якщо авторизації в системі не було, то користувачеві за замовчуванням присвоюється права для звичайного відвідувача. Реалізація функції «share» наведена в лістингу 3.14.

Лістинг 3.14 – Демонстрація функції «share»

```
public function share(Request $request): array
{
    return array_merge(parent::share($request), [
        'auth' => [
            'user' => $request->user() ?? [ //перевірка поточного
            'role' => User::ROLE_GUEST, // користувача
        ],
    ],
    ]);
}
```

Кінець лістингу 3.14

В компоненті «GuestLayout», виконується проста перевірка на те, яка саме роль в поточного користувача, якщо це адміністратор, то наступний елемент буде прихований, приклад наведено в лістингу 3.15.

Лістинг 3.15 – Демонстрація перевірки ролі користувача

```
{user.role == 0 && {/*перевірка ролі поточного користувача*/}
<LoginForm></LoginForm>
}
```

Кінець лістингу 3.15

Мінімальний функціонал адміністратора полягає в створенні та редагуванні даних про транспорт, тому було створено відповідні функції.

Функції зберігання та редагування подібні за своїм принципом. На сторінці, а саме в модальному вікні, є форма, яка відправляє дані до контролера, де вони перевіряються та зберігаються в базу даних. Для прикладу, вигляд одного з методів цього контролера наведено в лістингу 3.16.

Лістинг 3.16 – Демонстрація функції створення запису

```

public function store(Request $request)
{
    $data = $request->validate([ //валідація даних з форми
        'name' => 'required|string',
        'year' => 'required|string',
        'weight' => 'required|string',
        'ecology' => 'required|string',
        'capacity' => 'required|string',
        'type' => 'required|string',
        'routes' => 'required|string',
        'image' => 'nullable|file',
    ]);
    if ($request->file('image')) { //перевірка на наявність фото
        $data['image'] = $request->file('image')->store('uploads/',
'public');
    }
    Truck::Create($data);          //зберігання запису
    return to_route('truck');
}

```

Кінець лістингу 3.16

3.2 Тестування та налагодження інформаційно-комп'ютерної системи

Функціональне тестування – перевірка на те, чи правильно працюють ті чи інші функції розроблюваної системи. Під перевірку підпадають наступні елементи:

- авторизація адміністратора;
- створення, редагування, видалення записів для транспорту;
- відправка повідомлення через форму зворотнього зв'язку.

Авторизація для адміністратора, має працювати таким чином, що коли користувач, з відповідними правами, входить в систему, у нього пропадає кнопка аутентифікації та з'являється доступ до редагування записів транспорту. Для цього був створений тестовий запис в «users».

Першим кроком є перегляд сторінок без авторизації в системі. Доцільно лишитися саме на сторінці «Наш транспорт», тому що після входу адміністратора, тут з'явиться доступ до редагування записів (рис. 3.11).

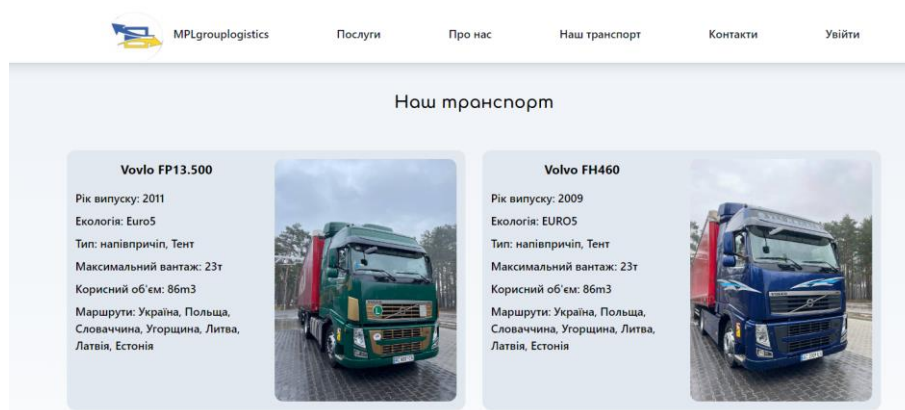


Рисунок 3.11 – Вигляд сторінки для відвідувача

Використовуючи попередньо створений запис адміністратора, потрібно увійти в систему. Форма авторизації з'являється після натискання кнопки «Увійти» (рис. 3.12).

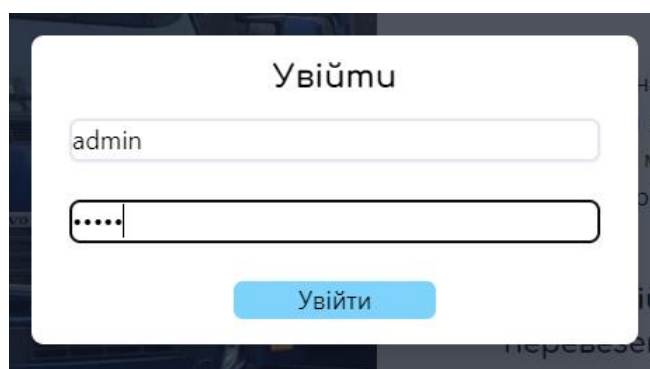


Рисунок 3.12 – Тестовий вхід адміністратора

Після входу в систему, з'явився функціонал адміністратора (рис. 3.13).

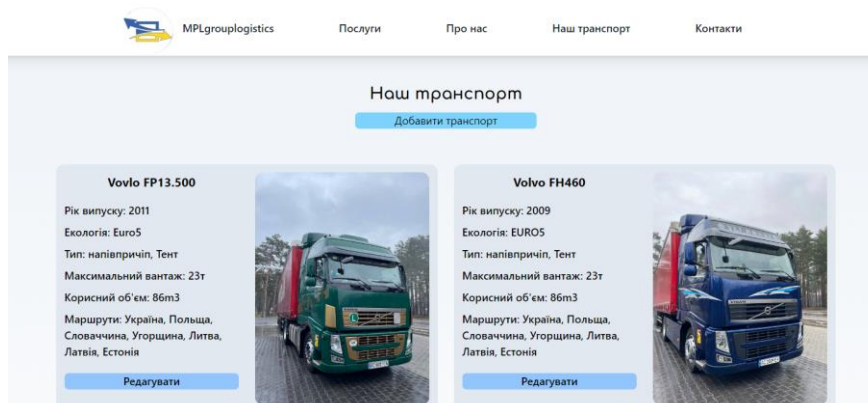


Рисунок 3.13 – Вигляд сторінки для адміністратора

Тестування створення запису транспорту відбувається після авторизації користувача з правами адміністратора. Спочатку потрібно відкрити модальне вікно, яке містить форму відправки даних, її було заповнено тестовими значеннями (рис. 3.14).

Рисунок 3.14 – Заповнена форма тестовими значеннями

Після натискання кнопки «Додати транспорт», дані було відправлено до серверної частини та збережено в базі даних, про це свідчить новий запис, який виводиться поряд з іншими (рис. 3.15).

Рисунок 3.15 – Вивід тестового запису

Тестування форми зворотнього зв'язку, починається з відкриття сторінки «Контакти» та наповнення її тестовими даними (рис. 3.16). Після натискання кнопки «Відправити», форма автоматично очищається.

Рисунок 3.16 – Заповнена форма зв'язку

Результатом успішної відправки даних, є лист, який прийшов на електронну пошту (рис. 3.17).

Рисунок 3.17 – Результат відправки даних

3.3 Розробка бази даних

Для розробки сайту-візитки, велика база даних не знадобиться. Для реалізації користувача з правами адміністратора та його можливістю змінювати дані для транспорту, потрібно три таблиці.

Таблиці «users» та «sessions», створюються при встановленні Laravel, вони відповідають за збереження даних про адміністратора.

Таблиця «trucks», має містити всю потрібну інформацію про автомобілі компанії, а саме поля: «name», «year», «weight», «capacity», «ecology», «type», «routes», «image».

При локальній розробці, таблиці створюються завдяки міграціям, які надає Laravel. Після їх створення, потрібно відредагувати поля та дозволити їх заповнення у моделі. Для прикладу, налаштований файл міграції для «trucks» наведено в лістингу 3.18.

Лістинг 3.18 – Демонстрація міграції для «trucks»

```
Schema::create('trucks', function (Blueprint $table) {
    $table->id();
    $table->string('name');
    $table->string('year');
    $table->string('weight');
    $table->string('capacity');
    $table->string('ecology');
    $table->string('type');
    $table->string('routes');
    $table->string('image')->nullable();
});
```

Кінець лістингу 3.18

При розгортанні розроблюваного сайту на хостинг, таблиці створювалися в phpMyAdmin через SQL запити. Загальний вигляд бази даних зображено на рисунку 3.18.

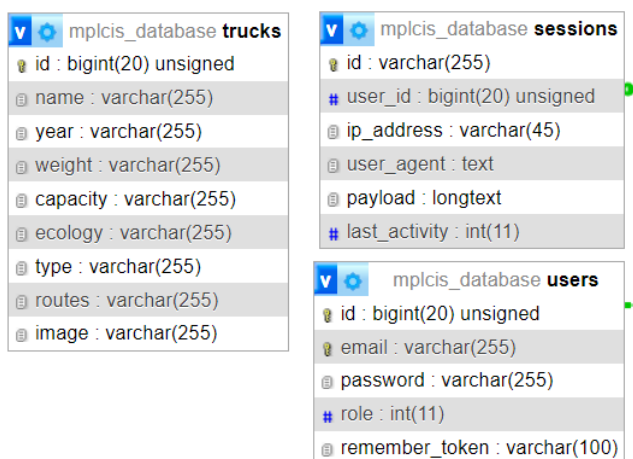


Рисунок 3.18 – Схема бази даних в phpMyAdmin

Для прикладу, SQL запит для створення таблиці «trucks» наведено в лістингу 3.19.

Лістинг 3.19 – Демонстрація SQL запиту створення таблиці «trucks»

```
CREATE TABLE `trucks` (
  `id` bigint unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `year` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `weight` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `capacity` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `ecology` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `type` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `routes` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `image` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE = InnoDB AUTO_INCREMENT = 13 DEFAULT CHARSET = utf8mb4 COLLATE = utf8mb4_unicode_ci
```

Кінець лістингу 3.19

3.4 Розгортання сайту на хостингу

Існує багато хостинг-провайдерів, кожен з яких має свої переваги та недоліки. В пріоритеті були ті хостинги, які пропонували безкоштовний план, можливість створення бази даних та кількість пам'яті, яку виділяє провайдер.

GoogieHost (рис. 3.19) – сервіс, який надає послуги хостингу, його безкоштовний план містить всі мінімально-необхідні ресурси для функціонування сайту (рис. 3.20).

Після вибору плану, надається вибір домена для сайту, найкращим варіантом є безкоштовний субдомен, який надає провайдер. Наступним кроком є створення акаунта, який буде використовуватись для адміністрування сайту.

Після створення акаунта, надається інформація про вхід в панель управління, де і будуть проводитись наступні встановлення та налаштування:

- встановлення останньої версії PHP;
- архівування всіх файлів та завантаження їх на хостинг;
- налаштування папки «Public_html»;

— створення та налаштування бази даних.



Рисунок 3.19 – Головна сторінка GoogieHost [16]



Рисунок 3.20 – Головна сторінка GoogieHost [16]

Для створення бази даних, на хостингу є phpMyAdmin. Використовуючи цей інструмент та налаштувавши файл «.env», було створено та підключено базу даних (рис. 3.21).

					Database Count	1 / 1	Total Size	96 KB
Database Name	Size	Users	Tables	Actions				
 mpicis_database	96 KB	1	3	Manage	Delete			

Рисунок 3.21 – Створена та підключена база даних

Сервіс також надає безкоштовний SSL сертифікат, але є певні обмеження в їх кількості для хостингу, незважаючи на це, його було додано (рис. 3.22). Розміщений сайт на хостингу, з доступом до інтернету, зображено на рисунку 3.23.

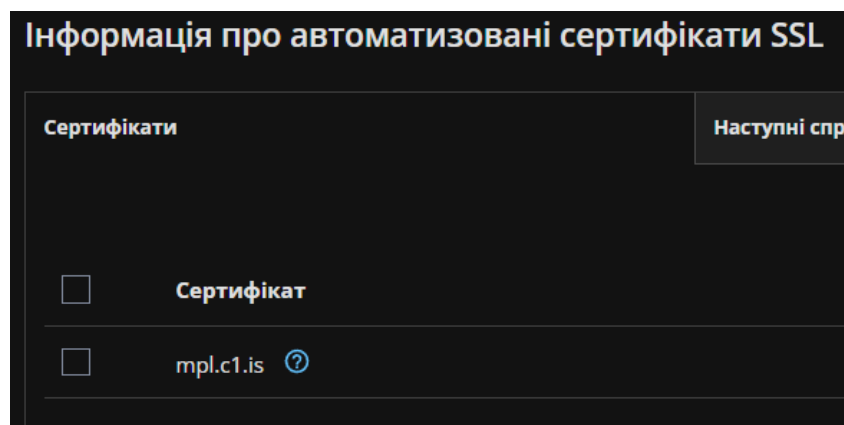


Рисунок 3.22 – Наданий SSL сертифікат

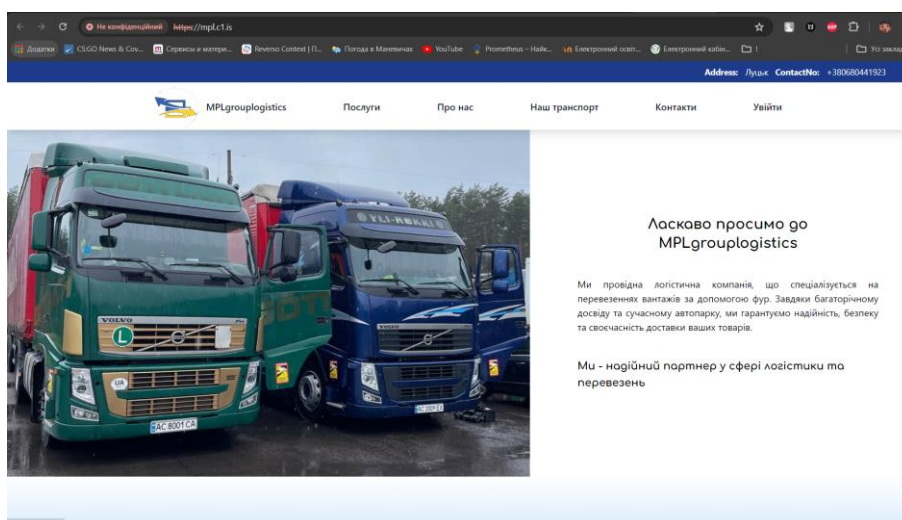


Рисунок 3.23 – Розміщений сайт на хостингу

3.5 Захист інформаційно-комп'ютерної системи

Для захисту розроблюваного сайту-візитки, присутні механізми авторизації та автентифікації. Впровадження таких функцій у веб-додатки є складним та ризикованим процесом, але фреймворк Laravel надає деякі інструменти, які спрощують їх реалізацію.

Автентифікація – перевірка користувача при вході в систему. В розроблюваному сайті, цей механізм можна описати наступним чином:

- адміністратор вводить свої дані в формі автентифікації;
- дані, у вигляді пароля та логіна, надходять до контролера;
- в контролері, дані проходять валідацію;
- на основі валідованих даних, система намагається автентифікувати адміністратора, при цьому порівнюючи дані з БД, де пароль зберігається в хешованому вигляді;
- якщо успішно, додатково створюється сесія сеансу та автентифікованого адміністратора перенаправляє на головну сторінку.

Авторизація – перевірка прав користувача та те що він може робити в системі. В розроблюваному сайті, за цей механізм захисту відповідає «Middleware», також можуть бути перевірки в клієнтській частині додатку. Він працює на звичайній перевірці ролі користувача, його наведено в лістингу 3.20.

Лістинг 3.20 – Демонстрація класу «AdminMiddleware»

```
class AdminMiddleware
{
    public function handle(Request $request, Closure $next): Response
    {
        if(Auth::user()->role !== User::ROLE_ADMIN){
            abort(404);
        }
        return $next($request);
    }
}
```

Кінець лістингу 3.20

Також клас «AdminMiddleware», потрібно підключити до маршрутизації тих шляхів, які повинні бути захищені від звичайного користувача, налаштування цієї частини наведено в лістингу 3.21.

Лістинг 3.21 – Маршрутизація, яка керується «AdminMiddleware»

```
Route::middleware('admin')->group(function () {
```

```
Route::post('/trucks/store', 'store');  
Route::post('/trucks/update/{truck}', 'update');  
Route::delete('/trucks/delete/{truck}', 'delete');  
});
```

Кінець лістингу 3.21

Висновки до розділу 3

У даному розділі, проведено розробку сайту-візитки для логістичної компанії «MPL». Перед початком створення, було налаштовано вибрані раніше технології, після чого почалась власне реалізація сайту.

Було створено front-end та back-end частини сайту, що забезпечують раніше описані функціональні вимоги до розроблюваного інтернет-ресурсу. Також, всі сторінки вміщують всю потрібну інформацію про компанію. В ході розробки, було під'єднано та налаштовано базу даних. Проведено функціональне тестування авторизації, зміни записів транспорту та відправку листів через форму зв'язку. Розроблений сайт-візитка був розміщений на хостингу.

ВИСНОВКИ

Результатом виконання кваліфікаційної роботи бакалавра є розроблений сайт-візитка для компанії з міжнародних перевезень «MPL», який буде її представленням в інтернеті та виконуватиме функції реклами, комунікації з клієнтами. В ході розробки використовувались популярні та сучасні фреймворки Laravel та React, завдяки цим технологіям, створення сайтів є досить швидким та відносно простим, також слід зазначити подальшу можливість до масштабування проєкту.

Під час виконання кваліфікаційної роботи бакалавра було визначено актуальність проблеми розробки сайту-візитки для логістичної компанії. Аналіз сучасного стану сфери логістики показав, що для сталого розвитку, підвищення конкурентоспроможності необхідно мати сайт, який допоможе компанії збільшити обсяг використання своїх послуг, вийти на нові ринки та поліпшити комунікацію з клієнтами.

Визначено функціональні вимоги до розроблюваного сайту-візитки, які включають перегляд інформації для звичайного користувача та можливість редагувати дані для адміністратора. Вони були в повній мірі виконані в процесі реалізації сайту.

Було створено функціональну структуру сайту-візитки, яка містить наступні сторінки: «Головна», «Послуги», «Про нас», «Наш транспорт», «Контакти». Для всіх елементів структури, попередньо було описано вміст та схему розробки, що передбачає навігацію та авторське право.

Проведено аналіз способів та методів їх реалізації для розробки веб-сайтів, основні – створення вручну (HTML, CSS, JavaScript), використання CMS або фреймворків.

На основі проведеного аналізу існуючих способів та технологій розробки веб-сайтів, було обрано створювати сайт-візитку за допомогою фреймворків. Порівнявши різні back-end фреймворки, для створення серверної частини буде використовуватись Laravel, додатково описано способи його використання та

інструменти, які він надає. Для front-end частини – React, наведено його переваги та недоліки.

За допомогою обраних технологій, Laravel та React, було реалізовано сайт-візитку для компанії з міжнародних перевезень «MPL». Описано більшість етапів реалізації веб-сайту, а саме створення та наповнення сторінок, їх логіки та маршрутизації. Також було підключено базу даних, яка виконує роль збереження інформації. Весь процес розробки виконувався відповідно до попередньо написаних функціональних вимог та структури сайту.

В ході реалізації сайту було проведено функціональне тестування окремих елементів системи, як-от авторизація для адміністратора, зміна записів транспорту, а також коректність роботи форми зворотнього зв'язку.

Після завершення розробки, був обраний безкоштовний хостинг, де буде розміщуватись розроблений сайт. Після завантаження всіх файлів проекту, створення та налаштування бази даних, сайт-візитка для компанії з міжнародних перевезень «MPL» отримав доступ в інтернет.

Загалом, розроблений сайт-візитка для компанії з міжнародних перевезень «MPL», виконує свої функції, а саме залучення нових клієнтів, які тепер завжди зможуть переглянути актуальну інформацію про компанію.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Райсвіх Я. Р. Принципи менеджменту і особливості логістики в економіці підприємства. Посвідчення УкрІНТЕІ № 55 від 21.01.2021р., 2021. С. 154-157.
2. Чимош К. С. Генезис поняття «транспортна логістика». Агросвіт, 2020. № 17-18. С. 119-122.
3. Сенік Л. Основні напрямки розвитку транспортної логістики України: проблеми та виклики. Матеріали VI Міжнародної студентської науково-технічної конференції «Природничі та гуманітарні науки. Актуальні питання», 2023. С. 317-318.
4. Томчук О., Головченко Я. Аналітична оцінка послуг вантажних перевезень в умовах воєнного стану. Економіка та суспільство, 2023. С. 53.
5. Дрозд А. Логістична стратегія в системі управління міжнародною компанією. Інноваційні процеси економічного та соціально-культурного розвитку: вітчизняний та зарубіжний досвід : тези доп. XVI Міжнар. наук.-практ. конф. молод. учених і студ. Тернопіль: ЗУНУ, 2023. С. 182-183.
6. Олешко Т. І., Бахорчук В. Моделювання бізнес-процесів за допомогою діаграми випадків використання. Вчені записки Таврійського національного університету імені В. І. Вернадського, 2020. Том 32. № 3. С. 132-140.
7. HTML – Wikipedia. URL: <https://en.wikipedia.org/wiki/HTML> (дата звернення: 13.03.2024).
8. CSS – Wikipedia. URL: <https://en.wikipedia.org/wiki/CSS> (дата звернення: 14.03.2024).
9. JavaScript – Wikipedia. URL: <https://en.wikipedia.org/wiki/JavaScript> (дата звернення: 15.03.2024).
10. What is CMS? The Complete Guide. URL: <http://surl.li/uobbx> (дата звернення: 17.03.2024).
11. Що таке фреймворк: пояснюємо простими словами. Веб-студія Brainlab. URL: <http://surl.li/uobcb> (дата звернення: 20.03.2024).

12. 10 Best Backend Frameworks in 2024 | Radixweb. URL: <http://surl.li/uobcf> (дата звернення: 23.03.2024).

13. Advantages and Disadvantages of Django for Web Development. URL: <http://surl.li/uobcn> (дата звернення: 24.03.2024).

14. Most Popular Backend Frameworks in 2024 [What to Choose]. URL: <http://surl.li/uobci> (дата звернення: 26.03.2024).

15. Laravel – The PHP Framework For Web Artisans. Installation – Laravel 11.x The PHP Framework For Web Artisans. URL: <https://laravel.com/docs/11.x> (дата звернення: 26.03.2024).

16. Free Web Hosting And Domain Name | SSL, Php, MySQL, No Ads. URL: <https://googiesthost.com/> (дата звернення: 30.05.2024).