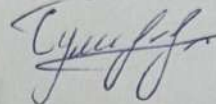


Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук

Бакалаврська кваліфікаційна робота на тему:

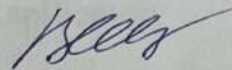
РОЗРОБКА WEB-РЕСУРСУ КОМПАНІЇ ІЗ ПЕРЕВЕЗЕННЯ ВАНТАЖІВ

Виконав: студент (4) курсу, групи ЗКН-216
спеціальності 122 – Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)



Гуменюк М. А.
(прізвище та ініціали)

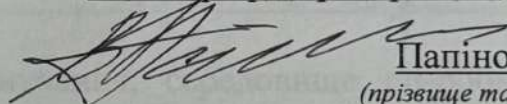
Керівник: к.т.н., доцент каф.КН



Денисюк В.О.
(прізвище та ініціали)

«04» червня 2025 р.

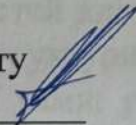
Рецензент: к.т.н, професор каф. АІТ



Папінов В.М.
(прізвище та ініціали)

«05» червня 2025 р.

Допущено до захисту

Зав. кафедри 

«06» червня 2025 р.

Вінниця ВНТУ - 2025 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти перший (бакалаврський)
Галузь знань – 12 Інформаційні технології
Спеціальність – 122 Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри КН

проф., д.т.н. Яровий А.А.

«05» травня 2025 р.

**ЗАВДАННЯ
НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**
Гуменюку Михайлу Андрійовичу

1. Тема роботи: Розробка WEB-ресурсу компанії із перевезення вантажів.

Керівник роботи: Денисюк Валерій Олександрович, к.т.н., доцент

затверджені наказом вищого навчального закладу «20» 03 2025 року № 97

2. Строк подання студентом роботи 30 травня 2025р.

3. Вихідні дані до роботи :

об'єктно-орієнтована мова програмування; середовище розробки Visual Studio; зручний інтерфейс; швидке завантаження сторінок; інформаційне наповнення; забезпечення статичного та динамічного наповнення сайту.

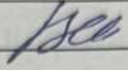
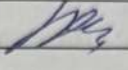
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

аналіз можливостей HTML; аналіз можливостей каскадних таблиць стилів CSS; аналіз можливостей клієнтської мови програмування JavaScript; постановка завдання; розробка структури сайту; реалізація web-сайтів; розробка інтерфейсу сайту; вибір колірної гами; розробка динамічного контенту; тестування роботи сайту.

5. Перелік графічного матеріалу:

схема алгоритму роботи модуля; діаграми класів та компонентів програмного забезпечення; загальний вигляд інтерфейсу користувача; приклад роботи програми.

6. Консультанти розділів проекту (роботи)

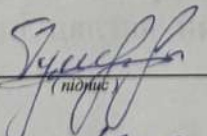
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Денисюк В.О., к.т.н., доцент		

7. Дата видачі завдання 5 травня 2025 року

КАЛЕНДАРНИЙ ПЛАН

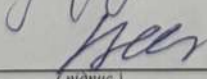
№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз предметної області WEB-ресурсу компанії із перевезення вантажів	05.05.25	07.05.25	
2	Постановка задачі	08.05.25	09.05.25	
3	Розробка програмного модуля WEB-ресурсу компанії із перевезення вантажів	12.05.25	15.05.25	
4	Обґрунтування засобів розробки WEB-ресурсу компанії із перевезення вантажів	14.05.25	16.05.25	
5	Програмна реалізація WEB-ресурсу компанії із перевезення вантажів	19.05.25	22.05.25	
6	Тестування WEB-ресурсу компанії із перевезення вантажів	30.05.25	01.06.25	
7	Розробка інструкції користувача	31.05.25	01.06.25	
8	Оформлення матеріалів до захисту БКР	02.06.25	04.06.25	

Студент


(підпис)

Гуменюк М.А.
(прізвище та ініціали)

Керівник роботи


(підпис)

Денисюк В.О.
(прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська кваліфікаційна робота складається з 75 сторінок формату А4, на яких є 18 рисунків, список використаних джерел містить 19 найменувань.

У роботі розглянуто актуальну задачу розробки WEB-ресурсу компанії із перевезення вантажів. Даний WEB-ресурс відповідає основним правилам побудови та наповненню графічною та текстовою інформацією. Статичний контент складається статичними та динамічними зображеннями різного формату. Сайт відповідає основним вимогам побудови та наповненню сайту графічною та текстовою інформацією. В процесі створення вебсайту було розроблено статичні та динамічні зображення різних форматів, при розробці використанні графічні редактори Gimp, Paint, Photoshop, Gif Animator, програмний пакет Notepad++ для написання HTML та CSS коду.

Ключові слова: перевезення вантажів, веб-додаток, Java, Photoshop, HTML, CSS.

ABSTRACT

The bachelor's qualification work consists of 75 pages of A4 format, on which there are 18 figures, the list of used sources contains 19 names.

The paper considers the current task of developing a WEB resource for a freight carrier company. This WEB resource meets the basic rules for building and filling with graphic and text information. Static content consists of static and dynamic images of various formats. The site meets the basic requirements for building and filling the site with graphic and text information. In the process of creating the website, static and dynamic images of various formats were developed, and the graphic editors Gimp, Paint, Photoshop, Gif Animator, and the Notepad++ software package for writing HTML and CSS code were used during development.

Keywords: freight transportation, web application, Java, Photoshop, HTML, CSS.

ЗМІСТ

ВСТУП.....	3
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	
КОМПАНІЇ ІЗ ПЕРЕВЕЗЕННЯ ВАНТАЖІВ.....	5
1.1 Аналіз галузі WEB-ресурсів.....	5
1.2 Аналіз відомих програм аналогів.....	6
1.3 Постановка задачі.....	11
1.4 Висновок до розділу 1.....	12
2 РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ WEB-РЕСУРСУ	
КОМПАНІЇ ІЗ ПЕРЕВЕЗЕННЯ ВАНТАЖІВ.....	13
2.1 Розробка структури WEB-ресурсу	
2.2 Розробка діаграми класів WEB-ресурсу	
компанії із перевезення вантажів.....	18
2.3 Послідовність дій WEB-ресурсу	
компанії із перевезення вантажів.....	19
2.4 Висновок до розділу 2.....	22
3 ПРОГРАМНА РЕАЛІЗАЦІЯ WEB-РЕСУРСУ	
КОМПАНІЇ ІЗ ПЕРЕВЕЗЕННЯ ВАНТАЖІВ.....	23
3.1 Обґрунтування вибору мови програмування.....	23
3.2 Створення WEB-ресурсу.....	30
3.3 Тестування WEB-ресурсу.....	44
3.4 Висновок до розділу 3	50
ВИСНОВКИ.....	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	53
ДОДАТОК А (ОБОВ'ЯЗКОВИЙ) ПРОТОКОЛ ПЕРЕВІРКИ	
КВАЛІФІКАЦІЙНОЇ РОБОТИ	55
ДОДАТОК Б (ОБОВ'ЯЗКОВИЙ) ЛІСТИНГ ПРОГРАМИ.....	56
ДОДАТОК В (ОБОВ'ЯЗКОВИЙ) ІЛЮСТРАТИВНА ЧАСТИНА.....	65
ДОДАТОК Г (ДОВІДНИКОВИЙ) ІНСТРУКЦІЯ КОРИСТУВАЧА.....	75

ВСТУП

Актуальність досліджень. Майже усі сучасні новітні технології пов'язані з розвитком комп'ютерних наук та інформаційних технологій. Інформаційні послуги, що надаються завдяки ним, можуть задовольнити майже усі верстви населення. Інформатизація сьогодні зумовлює розвиток держави, її безпеку та успішність.

У сучасному інформаційному просторі веб-ресурси стали важливим інструментом реалізації стратегій у сфері політики, економіки, соціального розвитку та національної безпеки. Динамічне зростання кількості нових інтернет-сайтів зумовлює потребу у створенні конкурентоспроможних та привабливих платформ, орієнтованих на інтереси цільової аудиторії.

Ключовими аспектами успішного веб-ресурсу є функціональність, унікальний дизайн, індивідуальність та здатність запам'ятовуватись. У контексті зростаючої конкуренції в онлайн-просторі важливим стає грамотний вибір концепції проєкту та залучення професійних виконавців. Незалежно від призначення сайту — презентація компанії чи монетизація через рекламу — його основним завданням залишається ефективно залучення широкої аудиторії.

Крім того, веб-сайти, пошукові системи та соціальні мережі виступають інструментами цифрової трансформації суспільства. Вони сприяють прискоренню доступу до інформації та розвитку новітніх технологічних рішень, що формують сучасний глобалізований світ.

Враховуючи сучасний розвиток промислової індустрії досить широко розвиваються вантажні перевезення, тому буде доцільним створення веб-сайту, який буде представляти користувачеві інформацію про компанію, що займається наданням послуг із вантажних перевезень. Отже, з вищенаведеного можна зробити висновок, що розробка WEB-ресурсу

компанії із перевезення вантажів є актуальною та потребує подальшого дослідження.

Метою дослідження є розширення функціональних можливостей WEB-ресурсу компанії із перевезення вантажів за рахунок реалізації дружнього та комфортного інтерфейса і збільшення галереї зображень.

Об'єктом дослідження є процеси, які виконуються WEB-ресурсом компанії із перевезення вантажів.

Предметом дослідження є алгоритми та програмне забезпечення, що втілює WEB-ресурс компанії із перевезення вантажів.

Задачі дослідження:

- дослідження вже наявних аналогічних рішень;
- вибір та обґрунтування підходу до вирішення поставленого завдання;
- розробка структури та дизайну WEB-ресурсу компанії із перевезення вантажів;
- програмне створення WEB-ресурсу компанії із перевезення вантажів;
- перевірка функціонування ресурсу та аналіз результатів тестування;
- створення користувацької інструкції з експлуатації WEB-ресурсу.

Апробація роботи.

Робота пройшла апробацію на конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2025), 2025р.», доповідь «Розробка сайту компанії перевізника вантажів.» [1].

Публікація: електронні тези конференції «Молодь у науці: дослідження, проблеми, перспективи (МН-2025), 2025р.» [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ КОМПАНІЇ ІЗ ПЕРЕВЕЗЕННЯ ВАНТАЖІВ

1.1 Аналіз галузі WEB-ресурсів

Особливості процесу розробки веб-ресурсу можна розглядати у такому контексті: етапи, вимоги та практичні аспекти.

Зовнішній вигляд веб-сайту, або його дизайн, відіграє ключову роль у формуванні першого враження користувача. Саме від якості візуального сприйняття значною мірою залежить, чи залишиться відвідувач на сайті, чи негайно його покине. Відтак, створення привабливого, зручного та впізнаваного дизайну є одним із головних завдань у процесі розробки інтернет-ресурсу.

Сайт як цифровий продукт являє собою сукупність HTML-сторінок, які функціонують під єдиним доменним ім'ям. Ефективне компонування, логічна структура та візуальна узгодженість усіх елементів — основа якісного веб-дизайну.

Процес створення дизайну зазвичай поділяється на кілька основних етапів.

Визначення мети та задач проєкту. На початковому етапі відбувається глибоке опрацювання цілей сайту, аналіз діяльності компанії-замовника, її ринку, конкурентного середовища та аудиторії. Формується базове бачення майбутнього ресурсу — його тип (інтернет-магазин, корпоративний сайт, портал, тощо) та приблизна структура.

Формування дизайнерської концепції. На цьому етапі визначаються основні розділи сайту, будується логіка навігації, обирається стильове рішення відповідно до попереднього аналізу. Важливим завданням є адаптація концепції під потреби цільової аудиторії.

Розробка та погодження макету. Візуальний макет сайту є прототипом,

що дозволяє замовнику ознайомитися з попереднім результатом. За необхідності вносяться корективи, що забезпечує максимальну відповідність очікуванням. Важливо, щоб коригування не перетворювалися на безкінечні зміни, адже це впливає на строки реалізації та бюджет.

Фіналізація дизайну та технічна реалізація. Після затвердження макету відбувається створення графічних елементів, верстка сторінок, налаштування функціоналу відповідно до затвердженої структури. На цьому етапі активно залучаються фахівці з фронтенд-розробки.

Окрему увагу слід приділяти кросбраузерності та адаптивності дизайну. Недостатня оптимізація під різні браузери або пристрої може негативно вплинути на вигляд ресурсу й загальне враження користувача. Наприклад, використання нестандартних шрифтів або невраховані технічні обмеження можуть спотворити зовнішній вигляд навіть найкращого дизайну.

Фінальним етапом є тестування готового сайту. Перевіряється працездатність усіх розділів, функціональних елементів, зручність користування (юзабіліті). Після запуску веб-ресурсу надзвичайно важливо забезпечити його технічну підтримку та регулярне оновлення, що дозволяє залишатися актуальним у динамічному онлайн-середовищі.

Таким чином, якісний веб-дизайн є не лише естетичним компонентом, а й стратегічним інструментом для залучення аудиторії та досягнення цілей компанії у цифровому просторі.

1.2 Аналіз відомих програм аналогів

Транспортні компанії зазвичай пропонують широкий спектр послуг: пасажирські та вантажні перевезення в межах міста, області, по всій країні та за кордон, міжнародні доставки, евакуація транспортних засобів, логістичне супроводження та інші експедиторські послуги [3-7]. Тип транспорту, який

використовується (автомобільний, залізничний, авіаційний, річковий чи морський), значною мірою визначає специфіку та масштаб діяльності компанії. Саме тому деякі транспортні оператори успішно поєднують кілька видів транспорту для забезпечення максимально ефективного перевезення. Також транспортна компанія може мати достатньо складну організаційну структуру (рис.1.1). Враховувати треба схему руху товарів, які перевозяться (рис.1.2). У свою чергу, операційна схема доставки товарів впливає на особливості розробки WEB-ресурсу компанії перевізника вантажів (рис.1.3).

Наявність власного вебсайту є важливим інструментом для компанії, що працює в сфері транспортних послуг. Такий ресурс значно полегшує процес просування бренду та залучення нових клієнтів, перебираючи на себе основні функції реклами.



Рисунок 1.1 – Розширена організаційна структура компанії

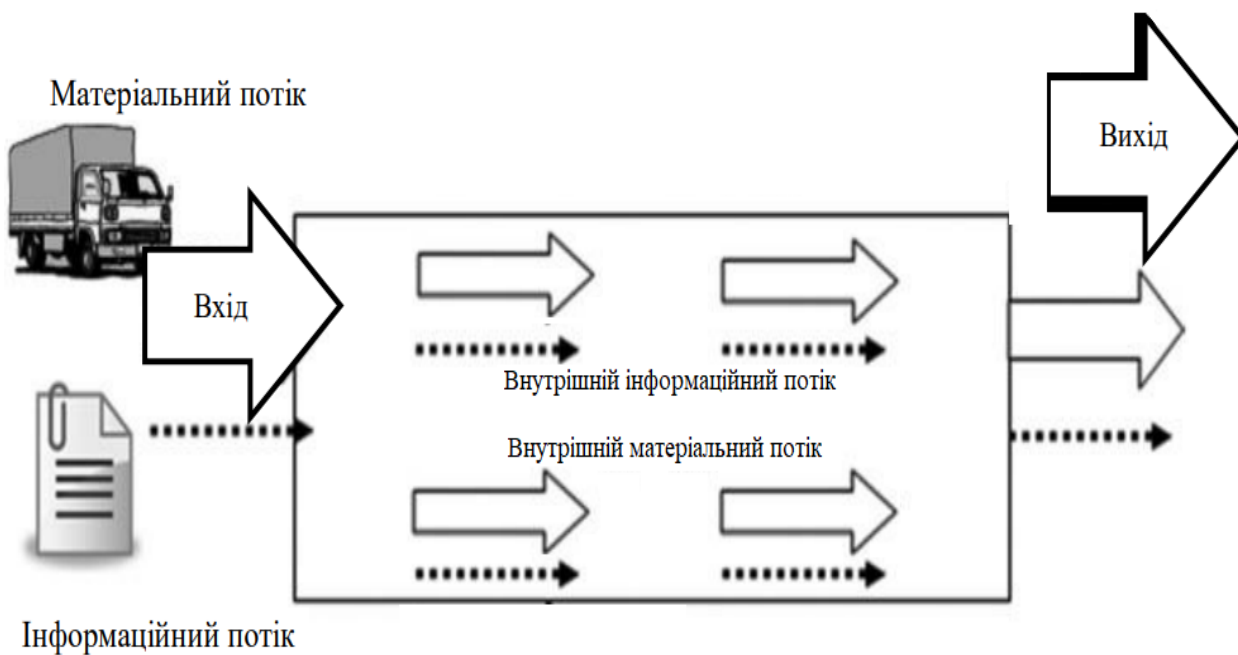


Рисунок 1.2 – Потокова схема транспортної системи

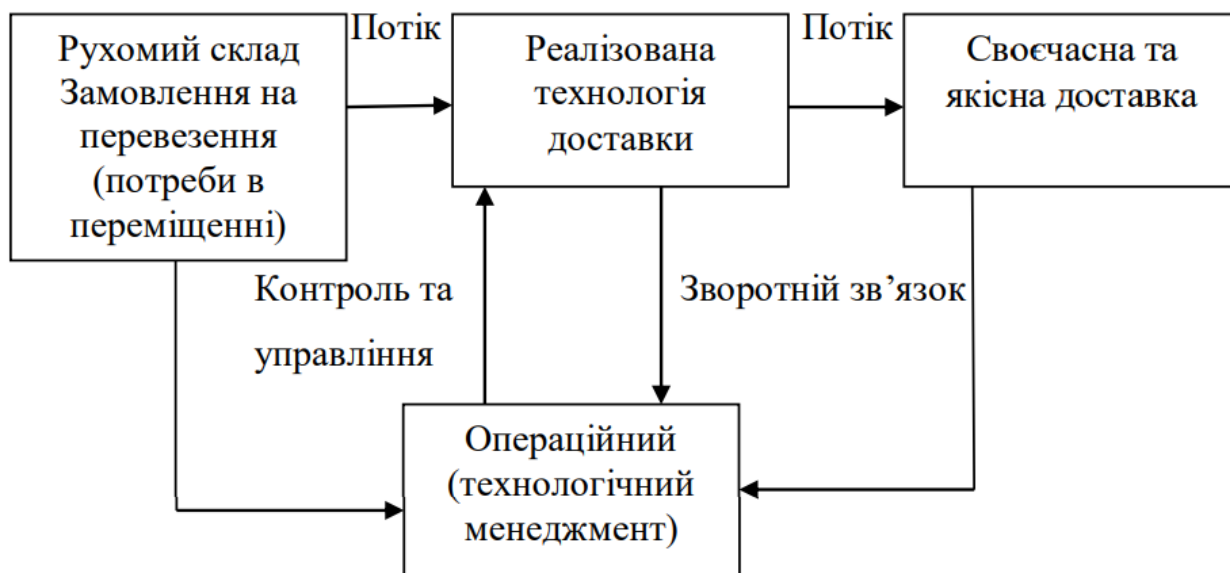


Рисунок 1.3 - Операційна схема доставки товарів

Крім того, інформація, представлена на сторінках сайту, допомагає зменшити навантаження на персонал, оскільки вже містить ключові

відомості, які цікавлять потенційних замовників. Нижче наведено приклад структурних розділів, які може включати сайт транспортної компанії:

- «Про нас» — загальна інформація про компанію, напрямки діяльності, досягнення та досвід роботи.
- «Пасажирські перевезення» — опис відповідних послуг, типів транспорту, графіків і цінової політики.
- «Вантажні перевезення» — деталі щодо перевезення вантажів, характеристика доступного транспорту, розцінки.
- «Графік руху» — розклад маршрутів, зокрема для пасажирських перевезень.
- «Онлайн-бронювання» — форма для замовлення квитків або подання заявки на перевезення вантажу.
- «Додаткові сервіси» — перелік супутніх послуг: доставка багажу, охорона вантажів тощо.
- «Оренда та продаж транспорту» — умови надання транспорту в оренду або його реалізації, вартість послуг.
- «Часті запитання» — відповіді на типові питання клієнтів щодо послуг компанії.
- «Відгуки» — коментарі та оцінки клієнтів, які вже скористалися послугами компанії.
- «Фотогалерея» — зображення транспортних засобів, фотозвіти з корпоративних або важливих подій.
- «Контакти» — адреса головного офісу, телефони, e-mail, інформація про регіональні представництва та карта з маршрутами.

Аналіз статистики запитів, пов'язаних з послугами компаній перевезення вантажів (табл.1.1), підтверджує важливість розробки WEB-ресурсу компанії перевізника вантажів [7].

Десятки тисяч запитів у пошукових системах щодня свідчать про значний інтерес користувачів до послуг транспортних компаній.

Якщо у перевізника є власний бізнес-сайт, він має всі шанси бути поміченим завдяки ефективним і водночас економним методам просування в інтернеті. Один із таких підходів — це пошукова оптимізація з урахуванням регіональної специфіки.

Геотаргетинг дозволяє налаштувати видимість сайту в результатах пошуку так, щоб він з'являвся переважно користувачам із конкретного регіону. Це значно підвищує шанси на залучення цільової аудиторії саме в потрібній географічній зоні.

Завдяки доступності у режимі 24/7, зручності перегляду, швидкому оновленню інформації та інтеграції з різноманітними онлайн-сервісами, вебсайт транспортної компанії стає не лише джерелом інформації, а й ефективним інструментом взаємодії з клієнтами. Компанія значно полегшує комунікацію та підвищує якість обслуговування, перетворюючись на справжнього цифрового помічника бізнесу.

Таблиця 1.1 – Статистика запитів, пов'язаних з послугами компаній перевезення вантажів

Пошуковий запит	Кількість запитів на місяць
Перевезення	463477
Транспортна компанія	157600
Перевезення вантажів	106385
Транспортні послуги	33480
Вантажні перевезення	32322
Пасажирські перевезення	28751
Міжнародні перевезення	25114
Транспортні перевезення	13376

1.3 Постановка задачі

Розглянемо формування завдань у процесі створення веб-сайту.

У процесі розробки веб-ресурсу визначення завдань виступає важливою складовою, яка безпосередньо впливає на досягнення загальної мети проєкту. Кожне завдання представляє собою окрему дію або комплекс дій, що у сукупності забезпечують ефективну реалізацію проєкту.

Формування переліку завдань здійснюється з урахуванням ряду чинників, серед яких ключову роль відіграють фінансові можливості, стратегічна значущість проєкту для організації, технічні та людські ресурси, доступні для його реалізації, а також часові рамки, у межах яких необхідно завершити розробку.

Коректна та чітка постановка завдань на початковому етапі проєктування дозволяє оптимізувати робочі процеси, уникнути непорозумінь між учасниками команди та забезпечити логічну послідовність виконання робіт. Саме тому даний етап вважається одним із найвідповідальніших у створенні веб-сайту.

Перед розробниками сайту, можуть ставитися різні завдання.

У межах розробки сучасного веб-ресурсу виділяються основні завдання, спрямовані на забезпечення його ефективності та відповідності потребам цільової аудиторії. До основних з них належать такі.

- Створення унікального дизайну, що гармонійно поєднується з корпоративною айдентикою та відображає візуальний стиль організації.
- Розробка корисних функціональних модулів, орієнтованих на аудиторію сайту: інформаційні блоки, пошукові системи, розважальний контент тощо.
- Інтеграція інтерактивних сервісів, що забезпечують зворотний зв'язок та взаємодію між користувачами: форуми, коментарі, блоги,

повідомлення, оголошення.

- Впровадження інструментів маркетингової взаємодії, таких як опитування, анкети, форми зворотного зв'язку та елементи тестування.
- Забезпечення надання онлайн-послуг, зокрема можливості електронної комерції (інтернет-продажі), систем бронювання, електронних платежів тощо.
- Створення зручних інструментів для керування сайтом. Перш за все, це система керування контентом, керування правами доступу до різних ресурсів сайту.
- Додаткові сервіси - взаємодія з бухгалтерським програмами, деякі елементи автоматизованих CRM (Customer Relationship Management) та ERM (Enterprise Relationship Management) систем. Як приклад можна навести систему розподілу заявок на доставку між кур'єрами в інтернет-магазині.

1.4 Висновок до розділу 1

Дослідження та розробка власного WEB-ресурсу компанії перевізника вантажів є актуальною задачею, розв'язання якої надає переваги у гнучкому індивідуальному дизайні, наявності тільки необхідних функцій, контролю трафіку, заощадженню коштів. У якості завдання обираємо розробку WEB-ресурсу компанії перевізника вантажів.

2 РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ WEB-РЕСУРСУ КОМПАНІЇ ІЗ ПЕРЕВЕЗЕННЯ ВАНТАЖІВ

2.1 Розробка структури WEB-ресурсу

Одним з найважливіших етапів розробки сайту є розробка його структури. Від цього етапу безпосередньо залежить, наскільки зручним сайт буде для користувачів. Навіть самий досконалий з естетичної, програмної та змістовної точки зору сайт не буде ефективний, якщо випустити з уваги можливості користування даним сайтом, що прийнято називати словом «Usability». Usability – це визначення зручності користування сайтом. Користувач не повинен думати про те за яким посиланням йому необхідно перейти, щоб знайти необхідну інформацію. Структура сайту складається з: схеми ієрархії сторінок, системи навігації і єдиної структури сторінок. Кінцевим результатом розробки структури сайту, є так звана модель сайту – діюча модель, яка не має ні дизайну, ні контенту, однак дає повне уявлення про те, яка буде ієрархія сторінок і як буде виглядати система навігації. Система навігації – один з найважливіших елементів сайту, якому при розробці структури приділяється особлива увага. У таку систему входять такі основні компоненти.

- Базова навігація, що визначає основні шляхи переміщення користувача по сайту. Сюди входить загальне меню, що присутнє на всіх сторінках сайту і посилання на головні сторінки розділів, а також внутрішня навігація кожного розділу і підрозділу.
- Функціональні елементи – форми реєстрації та авторизації, система контекстного пошуку, форми підписки на розсилки та повідомлення.
- Додаткові навігаційні елементи для переміщень в межах сторінки або між сусідніми сторінками [8].

Розробка структури сайту включає все, що стосується його вмісту. Дуже важливо заздалегідь спланувати все з урахуванням вимог користувача. Тільки уявивши себе звичайним користувачем, можна спроектувати гарну структуру.

Структура сайту компанії, що займається вантажними перевезеннями повинна бути такою, щоб користувач з легкістю міг знайти та переглянути необхідну йому інформацію. Також структуру сайту потрібно розробити так, щоб користувач завжди міг перейти на довільну сторінку із будь-якої сторінки. Вигляд сторінки сайту зображено на рисунку 2.1.

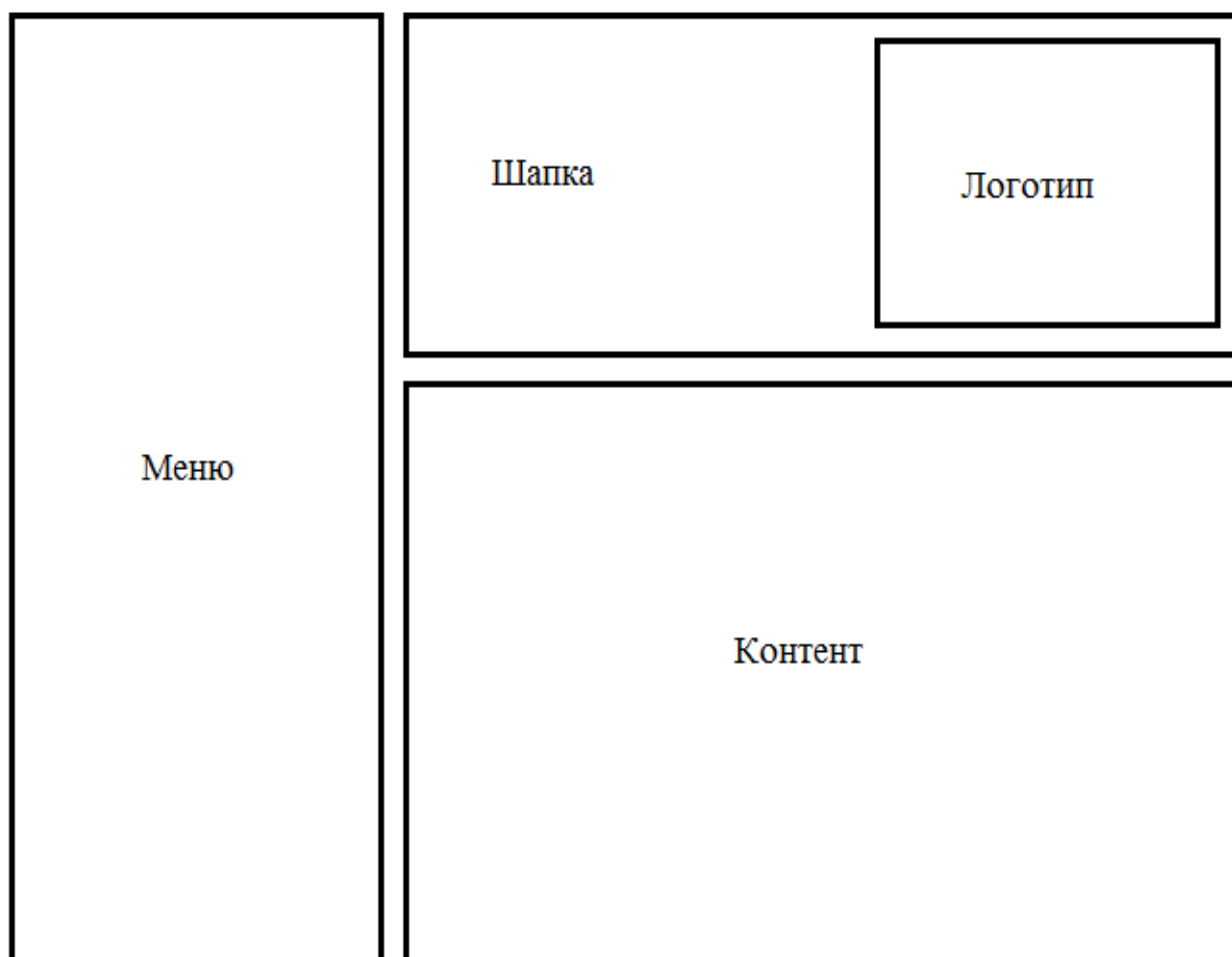


Рисунок 2.1 – Вигляд сторінки сайту

Архітектура веб-сайту умовно поділяється на два основні рівні: фізичний та логічний.

Фізичний рівень структури являє собою організований набір файлів різного формату — HTML-документів, графічних матеріалів, скриптів, медіаоб'єктів тощо. Грамотно побудована файлова система полегшує процес розробки, забезпечує ефективну взаємодію між членами команди та сприяє подальшій технічній підтримці ресурсу.

На логічному рівні сайт розглядається як система взаємопов'язаних веб-сторінок, що мають спільний візуальний стиль, структуру та єдину систему навігації. Саме логічна побудова забезпечує користувачу легкість у сприйнятті контенту та орієнтації на ресурсі.

Недостатньо продумана логічна структура негативно впливає на користувацький досвід: складність у пошуку потрібної інформації знижує рівень довіри до сайту та може призвести до втрати відвідувачів. Якщо користувач не може швидко знайти потрібний розділ або матеріал, він, як правило, припускає, що така інформація відсутня взагалі, і залишає ресурс.

Ефективний веб-дизайн повинен бути орієнтованим на кінцевого користувача. Важливо, щоб навігація була інтуїтивно зрозумілою, а логічна схема — прозорою та послідовною. Це передбачає чіткий зв'язок між сторінками, логічну організацію контенту і зручну структуру переходів.

Якісно реалізована навігаційна система дозволяє користувачеві чітко розуміти своє місцезнаходження на сайті, знати розміщення основних елементів та способи їх взаємодії. Крім того, добре побудована структура гарантує, що користувач зможе без труднощів повернутися до раніше переглянутої інформації, не втрачаючи контексту.

Розглянемо типи структур організації веб-сайтів.

У процесі проектування веб-ресурсів використовують різні моделі структурної побудови, кожна з яких має свої особливості застосування залежно від функціонального призначення сайту.

Лінійна структура передбачає послідовне розміщення сторінок, де навігація здійснюється шляхом переходу вперед або назад між суміжними сторінками. Така модель рідко застосовується в чистому вигляді через її обмежену зручність для кінцевого користувача. Найчастіше вона використовується для розбиття великого обсягу інформації на окремі, логічно відокремлені частини. Наприклад, під час публікації тексту книги сторінки можуть відповідати окремим главам.

Ієрархічна структура базується на побудові веб-сайту у вигляді дерева, де головна сторінка виконує роль центрального вузла з навігаційним меню. З неї користувачі переходять до основних розділів, які, у свою чергу, можуть містити підрозділи або додаткові сторінки з детальною інформацією. Така структура є зручною для систематизації великого обсягу контенту.

Мережна структура є найпоширенішою в сучасному веб-середовищі. Вона забезпечує гнучке переміщення між сторінками, що логічно пов'язані між собою, дозволяючи користувачеві переходити у будь-якому зручному для нього напрямку. Формування зв'язків у такій структурі відбувається відповідно до очікувань або поведінки відвідувачів. Типовими прикладами застосування є інтернет-магазини та багатофункціональні інформаційні портали.

Комбінована структура поєднує елементи кількох описаних вище типів. Наприклад, сайт може мати ієрархічну основу, в рамках якої реалізовані лінійні переходи для окремих процесів — таких як покрокове оформлення замовлення чи проходження реєстрації.

Навігація сайту є також важливим елементом. Хороша навігація дає відвідувачу можливість користуватися веб-сайтом просто, без усяких складностей і плутанини.

Розглянемо особливості побудови навігаційної структури веб-сайту.

Раціонально організована система навігації є ключовим чинником зручності використання веб-сайту. Вона забезпечує можливість безперешкодного переходу між різними сторінками ресурсу та гарантує користувачу впевненість у здатності швидко повернутися до раніше переглянутого розділу.

Варто підкреслити, що дизайн сайту розробляється насамперед з орієнтацією на потреби кінцевого користувача, а не з урахуванням виключно побажань власників або розробників ресурсу. Одним із головних принципів ефективної навігації є її логічна послідовність і внутрішня узгодженість.

Створення чіткої та несуперечливої навігаційної моделі дозволяє користувачам краще орієнтуватися в структурі сайту, розуміти своє поточне розташування та наявні функціональні можливості. Навігаційні елементи в цьому контексті виконують функцію орієнтирів — подібно до дорожніх знаків — які забезпечують інтуїтивно зрозуміле переміщення між розділами ресурсу та підвищують загальну ефективність користування сайтом.

На рисунку 2.2 надано обрану схему структури WEB-ресурсу компанії із перевезення вантажів, якої будемо притримуватися.

Короткий опис всіх складових схеми структури веб-сайту:

Index.html – головна сторінка веб-сайту на українській мові, яка містить загальну інформацію про вантажні перевезення;

About.html – сторінка веб-сайту на українській мові, що містить інформацію про компанію, яка займається вантажними перевезеннями;

Poslygi.html – сторінка веб-сайту на українській мові, яка містить інформацію про послуги, які надає компанія вантажних перевезень;

Galery.html – сторінка веб-сайту на українській мові, яка містить фотографії наявного транспорту для перевезення;

Contacts.html – сторінка веб-сайту українською мовою, яка містить інформацію компанію, що займається вантажними перевезеннями.

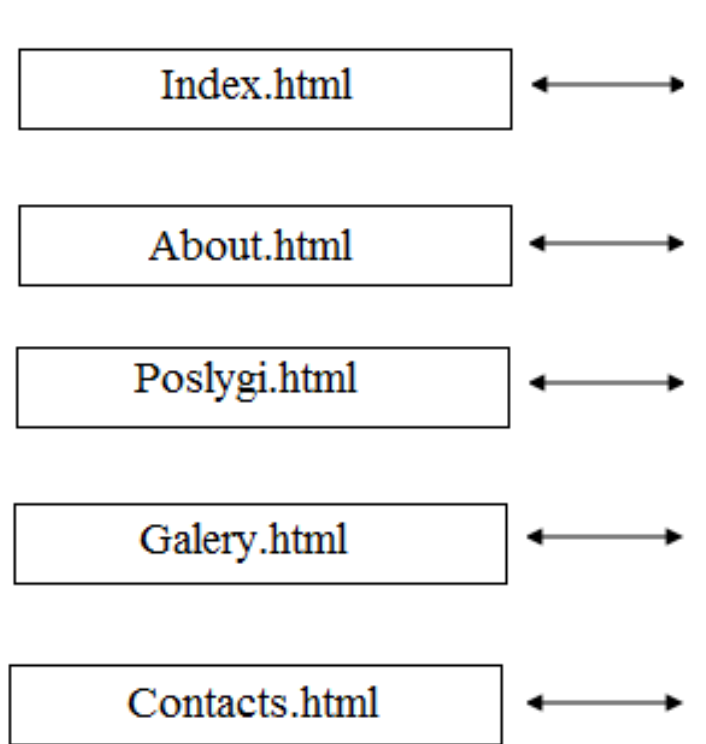


Рисунок 2.2 – Схема структури веб-сайту

2.2 Розробка діаграми класів WEB-ресурсу компанії із перевезення вантажів

UML-діаграма класів для сайту "Вантажні перевезення" надана на рисунку 2.3. Діаграма класів являється ключовим інструментом для моделювання структури об'єктно-орієнтованих систем. Вона дає змогу представити класи, їхні атрибути, методи та взаємозв'язки між ними.

Основні класи такі:

- TransportSite має у складі Header, Menu, Content, і можливо Footer;
- Menu містить список елементів типу MenuItem;
- Content поділений на блок новин і блок основного контенту

(NewsBlock, PageContent);

- PageContent може містити зображення (Image);
- NewsBlock містить новини типу News.

2.3 Послідовність дій WEB-ресурсу компанії із перевезення вантажів

Послідовність дій (UML-activity diagram) для програми сайту транспортної компанії, моделює послідовність дій користувача та системи під час основного сценарію використання, наприклад, перегляд інформації та оформлення заявки на вантажне перевезення (рис.2.4). Сценарій - користувач оформлює заявку на вантажне перевезення. Основні кроки такі.

1. Відкриття сайту.
2. Перехід у розділ "Вантажні перевезення".
3. Ознайомлення з інформацією про послуги.
4. Натискання кнопки "Замовити перевезення".
5. Заповнення форми (дані вантажу, маршрут, контакти).
6. Надсилання заявки.
7. Підтвердження успішного відправлення (відповідь системи).

Інструкція користувача, відповідно, містить такі кроки:

1. Початок
2. → "Відкрити сайт"
3. → "Перейти до розділу Вантажні перевезення"
4. → "Переглянути опис послуги"
5. → "Натиснути кнопку 'Замовити'"
6. → "Заповнити форму"
7. → "Надіслати заявку"
8. → "Показати повідомлення про успіх"
9. Кінець.

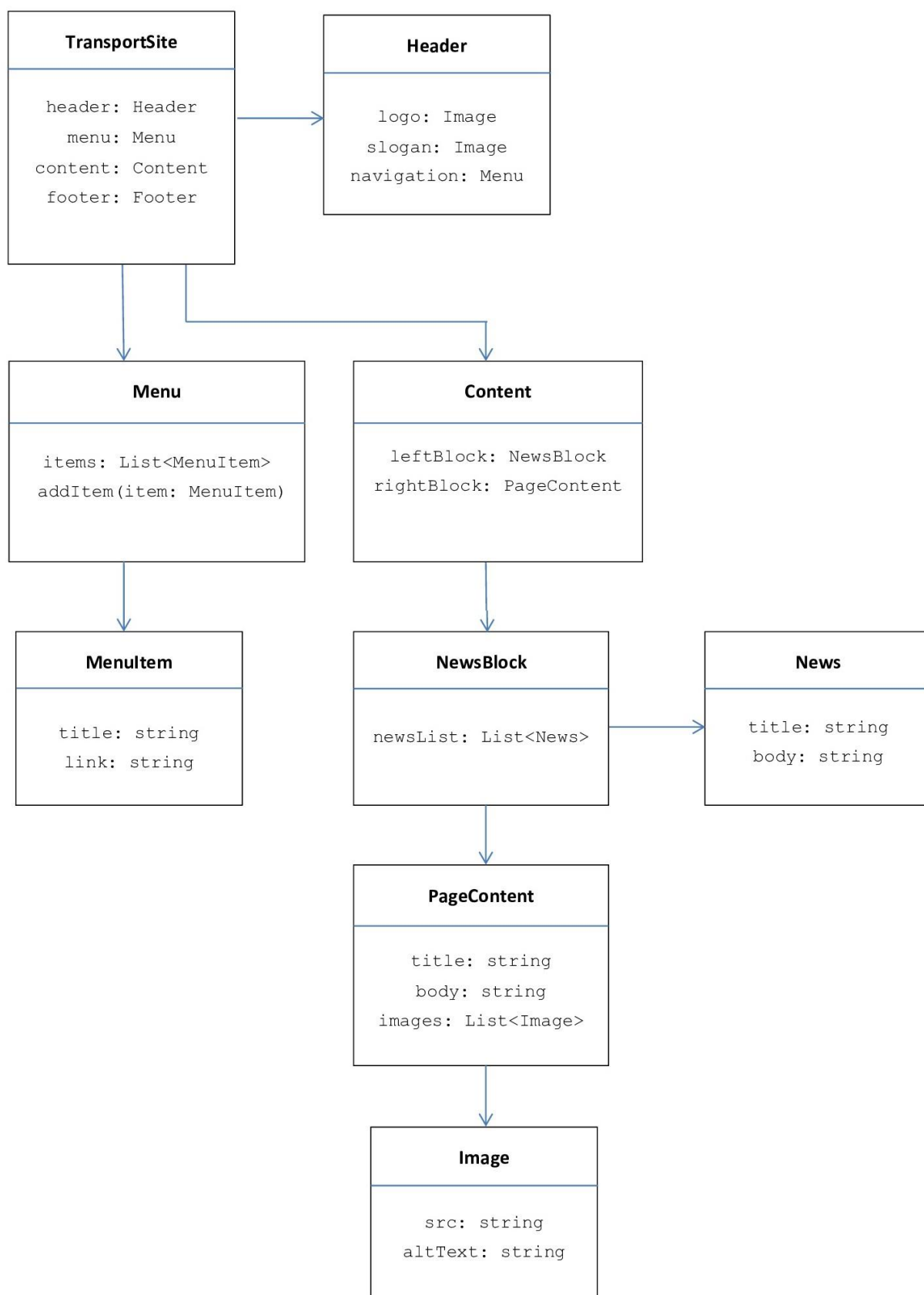


Рисунок 2.3 - Структура класів для сайту "Вантажні перевезення"

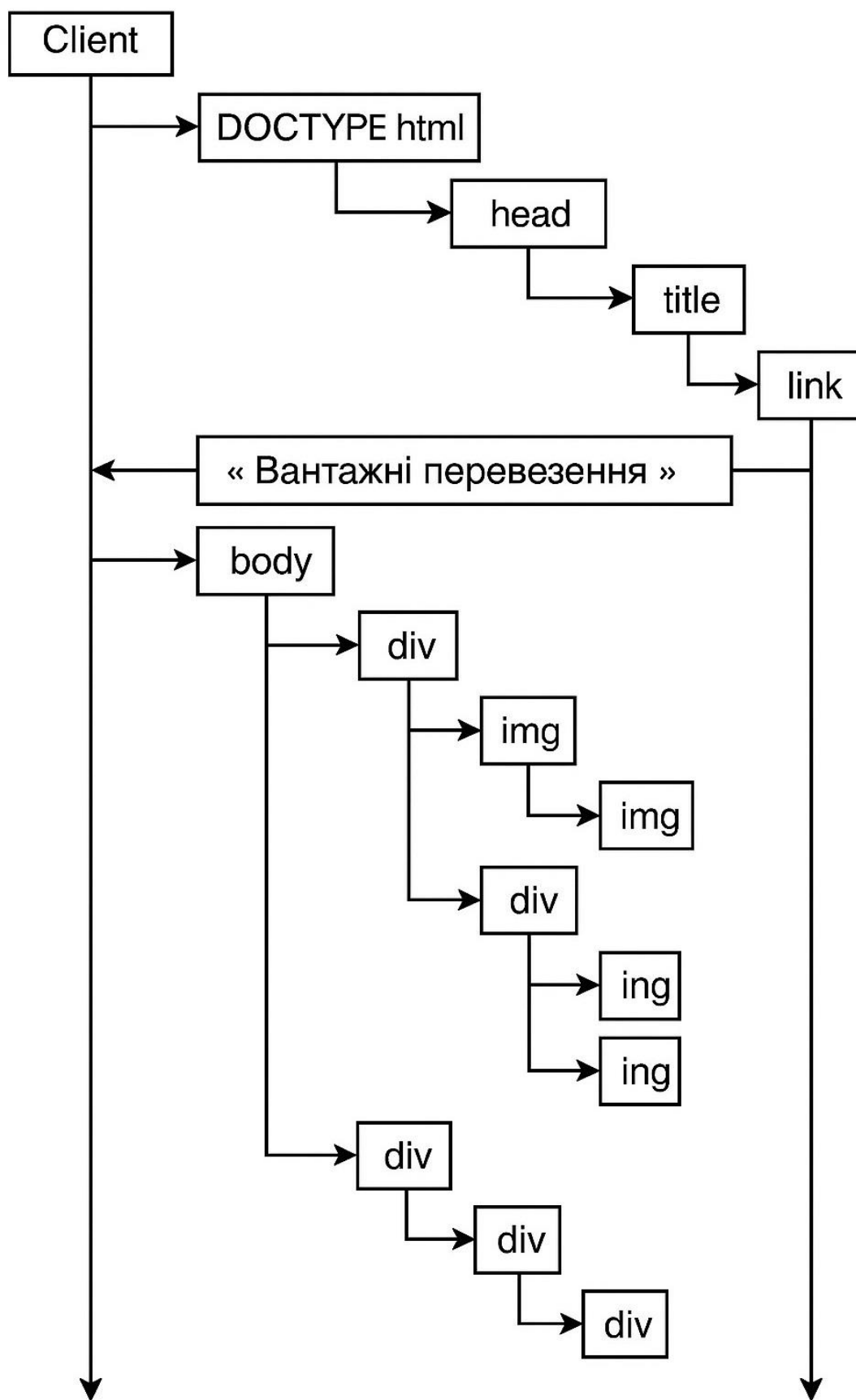


Рисунок 2.4 - UML-діаграма послідовності дій

UML-діаграма послідовності дій описує структуру інтерфейсу, стосується взаємодії користувача з інтерфейсом і завантаженням елементів (зображень, сторінок тощо) через браузер (рис.2.4).

Опис контексту.

HTML-сторінка — це веб-інтерфейс компанії вантажоперевезень. Вона містить:

- заголовок та меню;
- картинки;
- контент про діяльність компанії.

Основні учасники діаграми (Actors):

- користувач (User);
- браузер (Web Browser);
- HTML-документ;
- сервер (Web Server);
- зовнішні ресурси (зображення, CSS).

2.4 Висновок до розділу 2

У результаті досліджень та розробки обрана загальна структура WEB-ресурсу компанії із перевезення вантажів, визначено основні класи програмної реалізації сайту за допомогою діаграми класів, яка є ключовим інструментом для моделювання структури об'єктно-орієнтованих систем та дає змогу представити класи, їхні атрибути, методи і взаємозв'язки між ними, з'ясована послідовність дій WEB-ресурсу, яка моделює послідовність дій користувача та системи під час основного сценарію використання.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ WEB-РЕСУРСУ КОМПАНІЇ ІЗ ПЕРЕВЕЗЕННЯ ВАНТАЖІВ

3.1 Обґрунтування вибору мови програмування

Аналіз можливостей HTML.

HTML можна розглядати, як основу структури веб-сторінок. HTML (Hypertext Markup Language) — це спеціалізована мова розмітки, що слугує для визначення структури та представлення веб-документів. За допомогою HTML можна організовувати текстову інформацію у вигляді заголовків, абзаців, списків, таблиць, а також створювати гіперпосилання для навігації між окремими сторінками веб-простору. В основі HTML лежить текстовий формат, у якому елементи розмітки — теги — вбудовуються безпосередньо в код документа й слугують інструкціями для браузера щодо відображення вмісту на екрані.

Початкове формулювання HTML було розроблено Тімом Бернерсом-Лі в 1989 році як ключовий компонент для реалізації концепції розподіленої гіпертекстової інформаційної системи World Wide Web. Ідея гіпертексту передбачає вільне переміщення користувача між пов'язаними між собою документами без обов'язкового дотримання лінійної послідовності перегляду, характерної для друкованих видань.

HTML дає змогу створювати електронні документи з високим рівнем оформлення, що включають як текстові елементи, так і графічний, аудіо- та відеоконтент. Мова підтримує численні конструкції для організації інформації: різнорівневі заголовки, виділення тексту, структурування вмісту в логічні блоки тощо.

Однією з головних переваг HTML є його універсальність: створення HTML-документів можливе в будь-якому текстовому редакторі, незалежно від операційної системи чи апаратної платформи. Це зробило HTML

фундаментом веб-розробки й обумовило його широке розповсюдження як стандарту для побудови веб-сайтів [9].

Гіпертекстова організація даних та структура HTML-документів. У контексті технології World Wide Web, гіпертекстова база даних являє собою сукупність текстових документів, розмічених за допомогою мови HTML. Ця мова не лише визначає спосіб подання інформації на веб-сторінці, а й встановлює логічні зв'язки між різними документами за допомогою гіперпосилань.

Основою HTML є тегова модель, згідно з якою структура документа будується з використанням спеціальних команд — тегів. HTML-документ представляє собою звичайний текстовий файл у форматі ASCII, що містить інструкції форматування, позначені тегами. Кожен тег виконує певну функцію та зазвичай оформлюється у вигляді парного елемента — відкриваючого та закриваючого тегів, між якими розміщується вміст. Закриваючий тег позначається символом «/» перед ім'ям, наприклад, «</h1>».

Синтаксис HTML є достатньо інтуїтивним, оскільки базується на скороченнях англійських слів. Кожен тег може мати набір атрибутів, які визначають додаткові параметри його функціонування. Атрибути записуються після імені тегу, розділяються пробілами і приймають значення, що зазначаються після знака «=». Якщо значення складається з кількох слів, воно повинно бути взяте в лапки (одинарні або подвійні).

Деякі HTML-елементи, такі як тег зображення «», є самозакритими, тобто не вимагають парного закриваючого тегу. При створенні HTML-документів важливо дотримуватись правил вкладеності тегів — закривати їх у зворотному порядку до відкриття.

HTML-файли можуть бути створені у звичайних текстових редакторах, зокрема у стандартному «Блокноті» операційної системи Windows. Такий спосіб дозволяє безпосередньо керувати кодом і мінімізувати обсяг файлу. Альтернативою є використання візуальних HTML-редакторів, наприклад,

Microsoft FrontPage, які забезпечують графічний інтерфейс на зразок текстових процесорів (наприклад, MS Word). Проте слід враховувати, що документи, створені у візуальних редакторах, зазвичай мають більший обсяг, що може негативно вплинути на швидкість їх завантаження в умовах повільного з'єднання з Інтернетом.

Таким чином, мова HTML виступає ключовим інструментом для структурування, форматування та зв'язування вмісту веб-ресурсів, забезпечуючи функціональність та зручність користувацької взаємодії [9].

Аналіз можливостей каскадних таблиць стилів CSS.

Розглянемо каскадні таблиці стилів (CSS) та їх роль у веб-розробці [10]. Поняття таблиць стилів виникло ще в 1970-х роках у межах розвитку мови SGML, яка стала основою для сучасних мов розмітки. Зі зростанням популярності HTML виникла потреба у створенні ефективнішого механізму для визначення стилістичного оформлення веб-сторінок, що і стало передумовою до появи каскадних таблиць стилів — CSS (Cascading Style Sheets).

У процесі еволюції HTML у розробників з'явилися нові можливості для візуального оформлення елементів інтерфейсу, однак це ускладнило структуру коду та зменшило його гнучкість. Крім того, відмінності у відображенні веб-сторінок у різних браузерх створювали труднощі з уніфікацією зовнішнього вигляду, що знижувало контроль користувача над візуальним сприйняттям контенту.

CSS — це спеціалізована мова опису стилів, призначена для керування відображенням документів, написаних мовами розмітки, такими як HTML та XHTML. Її застосування дозволяє ефективно організовувати зовнішній вигляд веб-ресурсів — задавати шрифти, кольори, розміщення елементів, параметри відступів, а також адаптувати дизайн під різні типи пристроїв.

Розробка стандартів CSS здійснюється Консорціумом Всесвітньої павутини (W3C). Структура CSS поділяється на рівні (CSS1, CSS2, CSS3 тощо), де кожен наступний рівень базується на попередньому, додаючи нові функціональні можливості. Крім того, існують так звані профілі — специфічні варіанти CSS, призначені для певних пристроїв, наприклад, принтерів або мобільних телефонів.

Використання CSS дозволило перейти від табличної до блочної (каскадної) верстки, що забезпечує розділення змістової частини документа (контенту) та його зовнішнього оформлення. Такий підхід не лише спрощує процес розробки та підтримки веб-ресурсів, а й покращує їх адаптивність до різних середовищ відображення.

CSS активно застосовується як авторами веб-сторінок, так і кінцевими користувачами — для персоналізації інтерфейсу, зокрема шрифтів, кольорів і структури розміщення інформації. Однією з головних переваг CSS є можливість централізованого керування стилями для багатьох сторінок одночасно, що значно полегшує масштабну розробку веб-проектів [10].

Гнучкість відображення контенту за допомогою CSS. Використання каскадних таблиць стилів (CSS) дозволяє значно підвищити якість представлення інформації на веб-сторінках. Розділення контенту та його візуального оформлення сприяє кращому сприйняттю матеріалу, забезпечує зручність для користувачів із різними потребами та створює можливості для адаптивного дизайну. Такий підхід дозволяє усунути дублювання елементів оформлення, впорядкувати структуру документа, а також забезпечити контроль за відображенням у різних середовищах — від традиційних моніторів до мобільних пристроїв, принтерів, телевізорів, а також спеціалізованих засобів доступності, таких як брайлеві дисплеї або голосові браузері. HTML або XML-документи можуть мати різне візуальне представлення залежно від типу застосованих стилів.

Основні джерела стилістичних правил включають такі стилі [10].

- Браузерні стилі — стилі за замовчуванням, які застосовуються веб-оглядачем у разі відсутності або неповноти інших стилів.
- Користувацькі стилі — індивідуальні таблиці стилів, визначені користувачем у налаштуваннях браузера (наприклад, у браузері Opera).
- Вбудовані стилі — стилі, які задаються в ``<style>``-блоці в межах документа.
- Зовнішні стилі — окремі файли з розширенням ``.css``, які підключаються до HTML-документу.
- Авторські стилі — правила оформлення, створені безпосередньо розробником сторінки.
- Локальні стилі — оформлення, що задається безпосередньо в елементі за допомогою атрибута ``style``.

Завдяки каскадному механізму, стилі з різних джерел комбінуються відповідно до певних правил пріоритетності, що дозволяє досягати високої гнучкості в оформленні інтерфейсу веб-додатків та забезпечувати комфортне користування незалежно від пристрою чи умов перегляду.

Стандарт CSS визначає порядок та діапазон застосування стилів, те, в якій послідовності і для яких елементів застосовуються стилі. Таким чином, використовується принцип каскадності, коли для елементів вказується лише та інформація про стилі, що змінилася або не визначена загальнішими стилями.

Основні переваги використання CSS такі [10].

- Оптимізація продуктивності — використання CSS сприяє зменшенню розміру веб-сторінок та прискоренню їх завантаження. Сучасні веб-браузери зберігають CSS-файли у кеші, що дозволяє повторно використовувати ці стилі на всіх сторінках сайту без

необхідності повторного завантаження, знижуючи навантаження на сервер і мережу.

- Каскадні таблиці стилів (CSS) надають веб-розробникам цілий ряд функціональних переваг, які сприяють ефективному управлінню зовнішнім виглядом веб-сайтів та покращують загальну якість взаємодії з користувачем/
- Поліпшення структури HTML-документів — відокремлення стилів від основного вмісту дозволяє зменшити обсяг коду, підвищити його читабельність та забезпечити логічну організацію елементів сторінки відповідно до встановлених правил.
- Адаптація до потреб різних користувачів — CSS дає змогу створювати стилі, які враховують специфіку використання: наприклад, збільшений шрифт для людей з вадами зору, оптимізоване відображення для друку чи спеціальне оформлення для мобільних пристроїв.
- Централізоване керування оформленням — стилі для всього сайту або окремих його розділів можуть зберігатися в одному зовнішньому файлі з розширенням ``.css``. Це значно спрощує підтримку ресурсу, дозволяючи швидко вносити зміни в дизайн без потреби редагувати кожну сторінку окремо.

Аналіз можливостей клієнтської мови програмування JavaScript.

Мова JavaScript розглядається як мова сценаріїв для веб-розробки. JavaScript — це реалізація специфікації ECMAScript, створена компанією Netscape. Вона базується на концепції прототипного програмування та активно використовується для написання інтерактивних сценаріїв на веб-сторінках. Крім веб-розробки, JavaScript застосовується також у вбудованих рішеннях, де потрібне керування поведінкою елементів інтерфейсу.

Попри схожість назв, JavaScript не є спрощеною версією мови Java — це абсолютно різні технології. Торговельна марка "Java" належить компанії Sun Microsystems, і зв'язок між цими мовами є переважно маркетинговим.

JavaScript був впроваджений у браузер з метою динамічної взаємодії з HTML-документами. Його код інтегрується безпосередньо у сторінку, і обробляється вбудованим у браузер JavaScript-рушієм. Він дозволяє реагувати на дії користувача, обробляти введені дані, змінювати вміст сторінки без її повного перезавантаження тощо.

Хоча JavaScript вважається об'єктно-орієнтованою мовою, її модель роботи з об'єктами суттєво відрізняється від класичних ООП-підходів. Це зумовлено використанням механізму прототипів замість класів. Крім того, мова підтримує ряд функціональних можливостей, таких як замикання (closures), анонімні функції, каррінг, функції як об'єкти першого класу.

Основні риси мови такі:

- С-подібний синтаксис, який успадкований від С і спрощений у порівнянні з Java або C++;
- підтримка функцій вищого порядку;
- гнучке керування виключеннями — через конструкції `try...catch`;
- можливість створення анонімних функцій;
- автоматичне збирання сміття — непотрібні об'єкти видаляються автоматично;
- динамічна типізація — змінні не мають фіксованих типів, а значення можуть змінюватися у процесі виконання.

JavaScript має вбудовані об'єкти, які забезпечують базову функціональність, серед яких: `Object`, `Function`, `Array`, `Date`, `RegExp`, `Math`, `Error`, `String`, `Number`, `Boolean`, та інші. Разом із великою кількістю операторів та методів ці об'єкти формують основу для створення динамічних і функціональних веб-додатків [11].

Оскільки JavaScript є мовою інтерпретованого типу з динамічною типізацією та виконується у різних середовищах (браузерах, серверах, мобільних пристроях тощо), програмісту важливо враховувати відмінності в реалізації рушіїв JavaScript. Через відсутність суворої типізації та стандарту поведінки у всіх середовищах можуть виникати ситуації, коли скрипт, що працює коректно в одному браузері, дає інші результати або помилки в іншому.

Така мінливість виконання змушує розробника проводити додаткове тестування скриптів у кількох браузерах і на різних платформах, щоб гарантувати однакову функціональність. Це особливо актуально при створенні складної взаємодії між HTML-розміткою та JavaScript-функціональністю.

Оскільки кожен скриптовий блок обробляється окремо, хорошою практикою є винесення сценаріїв у зовнішні файли з розширенням `.js`, або створення окремих функціональних блоків JavaScript у межах сторінки. Це полегшує виявлення синтаксичних помилок, дозволяє швидше знаходити та ізолювати проблемні ділянки коду. Якщо код розміщено централізовано, помилки в одному фрагменті не зупиняють обробку всієї HTML-сторінки. До того ж, у разі помилки, можливо організувати обробку винятків таким чином, щоб забезпечити плавний вихід або підвантаження резервного сценарію без серйозного впливу на користувацький досвід [11].

3.2 Створення web-ресурсу

На першому етапі необхідно виділити найбільш важливі із завдань, реалізація котрих стоїть на перших місцях серед пріоритетів проекту. Всі завдання, які замовник ставить при розробці сайту, повинні відповідати загальній маркетинговій і рекламній політиці компанії.

Постановка завдання - дуже важливий етап у роботі над сайтом. Якщо завдання сформульовані правильно і чітко, навіть невдало вибраний розробник впорається з проектом. А головна помилка, яку допускають корпорації при створенні сайту, недооцінка тієї доданої вартості, яка закладається на цьому етапі.

Для написання html-коду можна використовувати велику кількість різних програмних пакетів, такі як: Adobe «Dreamweaver CS3», «Notepad++», «FrontPage» та інші.

На сьогоднішній день для створення веб-ресурсів використовують два основних типи редакторів: інструменти, що дозволяють працювати безпосередньо з програмним кодом, та візуальні редактори, які забезпечують створення сторінок без глибокого занурення в HTML. Обидва підходи мають свої переваги та обмеження.

У випадку використання текстових (кодових) редакторів, таких як Notepad++ чи Sublime Text, розробнику необхідно володіти знаннями мов розмітки, зокрема HTML і CSS. Такий метод дає повний контроль над структурою та стилем сторінки, але вимагає більшої технічної підготовки.

Візуальні редактори, як-от Microsoft FrontPage, працюють за принципом WYSIWYG (What You See Is What You Get), що дозволяє створювати веб-сторінки у графічному інтерфейсі, подібному до текстових редакторів, таких як Microsoft Word. Це значно спрощує розробку для початківців. До того ж, у таких програмах зазвичай передбачена можливість перегляду результату безпосередньо в робочому середовищі, що дозволяє швидко оцінити вигляд сторінки у веб-браузері без потреби в зовнішніх інструментах.

Microsoft FrontPage — це програмне забезпечення, призначене для створення веб-сайтів та окремих веб-сторінок. За допомогою меню та панелей інструментів користувач має змогу додавати на сторінку текстовий і графічний контент, елементи навігації, а також гіперпосилання для переходів між сторінками. Для візуального оформлення можна використовувати

таблиці, фрейми, анімаційні ефекти, біжучі рядки та інші елементи дизайну. Щоб полегшити процес створення веб-ресурсів, передбачено використання шаблонів і майстрів. FrontPage також забезпечує інтеграцію з іншими програмами пакета Microsoft Office, що дозволяє додавати вміст, підготовлений у Word, Excel або PowerPoint.

Для розробки веб-сайту «Компанії, що займається перевезеннями вантажів» обрано програмний пакет «Notepad++». Ця програма створена з доступним та простим для розробника інтерфейсом і в свою чергу вона є достатньо простою у використанні. Проте в програмі немає автоматичних підказок для написання того чи іншого тегу, під час написання коду веб-сторінки, що певним чином ускладнює роботу розробника.

Також задачею є розробка веб – сторінки з представленням інформації графічно. Розробка сторінки полягає в розробці дизайну, кольорової гами, та правильній комбінації графічної та текстової інформації, які несуть основну інформаційну частину сторінки.

Проектування є початковим і одним із ключових етапів створення веб-сайту. Для чого потрібен проект? Оскільки до розробки залучені фахівці з різних галузей, саме проект дозволяє кожному з учасників чітко усвідомити свої завдання та обсяг роботи. Підготовка проекту передбачає тісну співпрацю між замовником і розробником. Лише після завершення цього етапу можливо перейти безпосередньо до реалізації сайту [12].

Доменне ім'я. Це унікальна адреса веб-сайту, що складається з літер і цифр, за якою користувачі можуть знайти сайт у мережі, ввівши її в адресний рядок браузера. Домени класифікують на географічні (наприклад: .ru, .ua, .it, .us) та тематичні (.com, .org, .net). Ім'я повинно бути коротким, легко запам'ятовуватись і відповідати назві компанії, послуги або продукту. Якщо організація працює в межах певного регіону чи країни, доцільно обрати доменну зону, що відповідає географії її діяльності

Програмування. Цей етап є одним із найважливіших і найскладніших у процесі розробки веб-проекту. Від якості програмного коду залежить стабільність роботи сайту та його ефективна індексація пошуковими системами. Тривалість цього етапу визначається складністю технічного завдання та обсягом функціоналу: чим більше можливостей передбачено, тим складніше реалізувати проєкт. Якість програмування безпосередньо пов'язана з кваліфікацією фахівця.

Дизайн сайту. Під час створення візуального оформлення веб-ресурсу необхідно враховувати велику кількість аспектів. Зокрема, важливо правильно розмістити елементи керування, спираючись на принципи зручності користування (юзабіліті) та технічні вимоги, що забезпечують ефективне функціонування сайту. Веб-дизайн має свої унікальні особливості й суттєво відрізняється від інших видів графічного оформлення. Для створення якісного інтерфейсу необхідно мати знання мов веб-розробки та основ просування сайтів у пошукових системах. Лише з урахуванням цих факторів можна досягти стабільної та успішної роботи ресурсу [12].

Хостинг. Веб-сайт розміщується на спеціалізованому серверному обладнанні — це може бути один або кілька віддалених комп'ютерів. Для стабільного функціонування ресурсу необхідний цілодобовий доступ до нього, що потребує постійного з'єднання з Інтернетом, налаштування відповідного програмного забезпечення та технічної підтримки. Оскільки забезпечити це самостійно досить складно, цю послугу зазвичай надають хостинг-провайдери, які здають у користування частину дискового простору своїх серверів. Якість обраного хостингу безпосередньо впливає на доступність сайту, тому економити на ньому не варто. Доцільно оплачувати хостинг разом із доменом хоча б на рік наперед.

Просування ресурсу. Щоб веб-сайт був помітним для потенційних користувачів, його потрібно зареєструвати в пошукових системах та онлайн-каталогах. Однак через високу конкуренцію цього недостатньо — необхідно

комплексне SEO-просування. Дослідження показують, що користувачі здебільшого переглядають лише першу сторінку результатів пошуку. Тому для ефективної роботи сайту важливо забезпечити його присутність серед перших десяти результатів. Навіть якісна SEO-оптимізація не завжди гарантує високі позиції, тому рекомендується додатково використовувати інструменти інтернет-маркетингу.

Контент сайту. Один із найважливіших етапів — наповнення веб-ресурсу. Саме інформаційне наповнення бачать відвідувачі та індексують пошукові системи. Контент поділяється на текстовий — який повинен бути унікальним і адаптованим під пошукові запити (SEO), та графічний — зображення і відео, які візуально доповнюють інформацію й підвищують привабливість ресурсу. Основними вимогами до контенту залишаються висока якість та унікальність.

Супровід сайту. Після запуску веб-сайту важливо забезпечити його подальше обслуговування. Обсяг технічної підтримки залежить від типу ресурсу. Наприклад, сайт-візитка потребує мінімального втручання, оскільки інформація на ньому оновлюється рідко. Натомість інтернет-магазин вимагає постійного оновлення товарів, підтримки працездатності й SEO-позицій. Технічна підтримка може також включати в себе оновлення дизайну чи функціоналу [12].

Розглянемо розробку інтерфейсу сайту.

Веб-інтерфейс — це система засобів, яка дозволяє користувачам взаємодіяти з веб-сайтом або веб-додатком через браузер. Зі зростанням популярності Інтернету та широким розповсюдженням веб-браузерів такі інтерфейси набули особливої актуальності [13].

Найпоширенішим підходом до створення веб-інтерфейсів є поєднання HTML, CSS і JavaScript. Зазвичай це супроводжується використанням серверних скриптів. Однак варто враховувати, що через різне трактування стандартів HTML, CSS і DOM у різних браузерах можуть виникати труднощі

під час реалізації та підтримки веб-застосунків. Крім того, користувачі мають змогу змінювати налаштування свого браузера (наприклад, розміри шрифтів, кольорову схему чи вимикати виконання скриптів), що також може негативно вплинути на функціональність інтерфейсу.

Завдяки своїй доступності веб-інтерфейси є ефективним інструментом для організації віддаленої співпраці. Наприклад, вони активно використовуються в медіа, для ведення баз даних або спільної роботи над контентом. Класичним прикладом такої взаємодії є Wikipedia, де майже весь контент створюється та редагується через веб-інтерфейс.

Альтернативним, хоч і менш універсальним підходом до реалізації інтерфейсу є використання таких технологій, як Adobe Flash, Microsoft Silverlight або Java-аплети. Ці засоби часто функціонують через встановлення спеціальних плагінів, дозволяючи розробникам отримати більший контроль над поведінкою інтерфейсу. Утім, у таких рішеннях також можуть виникати проблеми сумісності на стороні клієнта.

Із розвитком DHTML і JavaScript значного поширення набув підхід, відомий як AJAX. Ця технологія дозволяє динамічно обмінюватися даними з сервером без необхідності повного перезавантаження сторінки, що значно підвищує інтерактивність і швидкість роботи веб-додатків.

Однією з ключових вимог до веб-інтерфейсів є їхня стабільна робота та однаковий зовнішній вигляд у різних браузерах. Це забезпечує користувачам єдиний досвід взаємодії незалежно від використовуваного програмного забезпечення.

Веб-інтерфейси також забезпечують доступ до різноманітних онлайн-сервісів і пристроїв з будь-якого місця. Але саме через цю відкритість особливу увагу потрібно приділяти питанням безпеки: автентифікації та авторизації користувачів, захисту переданих даних за допомогою шифрування, а також управлінню контентом і його модерації [13].

Дизайн веб-сайту часто ототожнюють із процесом створення інтерфейсу, через який користувач взаємодіє з контентом ресурсу. При розробці інтерфейсу важливо враховувати сприйняття та поведінку потенційних відвідувачів, адже саме від цього залежить ефективність сайту. Інтерфейси можуть варіюватися від дуже незручних до надзвичайно вдалих — усе залежить від підходу до проектування.

Часто найзручніші інтерфейси можна зустріти на сайтах інтернет-магазинів, де кожен елемент дизайну продуманий для залучення максимальної кількості користувачів. Для створення ефективного інтерфейсу достатньо дотримуватись основних принципів юзабіліті та мати естетичне чуття.

Один із ключових принципів — уникати зайвої складності. Надмірно складні або перевантажені рішення, які є спробою творчого самовираження дизайнера, можуть не знайти відгуку у користувачів і призвести до зниження відвідуваності. Тому доцільно дотримуватись знайомих шаблонів, які відповідають очікуванням аудиторії.

Інтерфейс повинен бути логічним, інтуїтивним та орієнтованим на основні потреби користувачів — інформування, взаємодію чи розвагу. Для цього на етапі планування команда проводить аналіз цільової аудиторії. Однак повне розуміння очікувань відвідувачів — непросте завдання: ні опитування, ні аналітика не дають вичерпної картини. Саме тому важливо поєднувати дослідження з практичним тестуванням та поступовим удосконаленням інтерфейсу [14, 15].

Принципи ефективного інтерфейсу веб-сайту. Одним із основних принципів, що працює як у розробці сайтів, так і в багатьох інших сферах — це простота. Надмірне ускладнення інтерфейсу лише шкодить користувацькому досвіду. Замість спроб охопити всі можливі функції й елементи одночасно, доцільно зосередитися на основних завданнях ресурсу та логічно їх реалізувати. Перевантаження сторінки великою кількістю

кнопок і посилань може дезорієнтувати користувача і відштовхнути його від взаємодії з сайтом.

Інтерфейс повинен бути спрямованим виключно на ті функції, які відповідають головній меті ресурсу. Користувачі, як правило, не читають увесь текст і не переглядають кожне зображення. Тому важливо чітко сформулювати пріоритети, структурувати навігацію та створити логічну карту сайту із зрозумілою системою переходів між сторінками.

Крім того, необхідно враховувати тип веб-ресурсу, що створюється. Наприклад, іміджеві чи промо-сайти орієнтовані на вузьку, але цільову аудиторію, тому їх інтерфейс може містити оригінальні дизайнерські рішення та елементи емоційного залучення. Водночас порталні сайти чи платформи з великою кількістю відвідувачів потребують максимально зручної, швидкої і простої навігації.

У будь-якому випадку, незалежно від стилю чи функціоналу, сайт повинен із перших секунд чітко демонструвати своє призначення. Користувач має одразу зрозуміти, де він знаходиться і яку інформацію або послугу може тут отримати.

Хороший інтерфейс неможливо налагодити без тестування. Знайдіть тестера з боку, дайте йому кілька завдань, наприклад: зареєструватися, знайти сторінку, знайти інформацію. Але до думки цього тестера варто прислухатися уважно, таким чином за одну подібну перевірку можна позбутися половини випадково допущених помилок [14, 15].

Розглянемо вибір колірної гами сайту.

Кольорова гама відіграє велике значення при розробці сайту, а особливо при розробці його дизайну. Кольори, що нас оточують в житті зазвичай, мають невелику насиченість і складаються з декількох відтінків, зате вони приємніші для очей, ніж відтворені чисті кольори. При використанні природних кольорів користувачі мають можливість краще зосередитися на самому сайті і не відволікатися через надто яскраві відтінки.

Адже відомо, що зосереджуючись на інформації, наші очі повинні бачити тільки цю картину, без сторонніх подразників [16]. Коли на сайті присутні яскраві відтінки кольорів, повністю зосередитись на чомусь одному неможливо.

Вплив кольорової гами на веб-дизайн. Вибір кольорів у процесі створення веб-сайту відіграє важливу роль, оскільки він безпосередньо впливає на емоційне сприйняття сторінки та поведінку користувачів. Щоб забезпечити ефективне візуальне оформлення, необхідно враховувати психологію кольору та його символіку.

Значення окремих кольорів.

- Білий — символізує чистоту, мінімалізм і гармонію. Дає змогу створити візуально легкий простір і підкреслити інші кольори на сторінці. Його доречно використовувати в сучасному дизайні, незважаючи на хибну думку про "нудність" білого фону.
- Оранжевий — теплий і енергійний, привертає увагу без агресії. Ідеальний для закликів до дії або виділення ключових елементів.
- Жовтий — виразний і життєрадісний, але його потрібно використовувати обережно, оскільки надлишок може викликати подразнення чи втому.
- Коричневий — природний, асоціюється з надійністю та простотою, хоча може виглядати надто стримано. Часто комбінується з яскравішими відтінками для досягнення балансу.
- Сірий — символізує професіоналізм, нейтральність і виваженість. Добре підходить для фонів або елементів, що не повинні привертати надмірної уваги.
- Червоний — асоціюється з енергією, пристрастю, динамікою та терміновістю. Він може викликати емоційний відгук і часто застосовується для акцентів, що стимулюють до дії.

Фактори, які слід враховувати при виборі кольорової палітри.

- Доступність: кольори слід підбирати з урахуванням потреб людей із вадами зору, забезпечуючи читабельність незалежно від особливостей сприйняття.
- Візуальна виразність: елементи, що потребують уваги, мають бути помітними та логічно виокремленими.
- Зручність читання: важливо забезпечити достатній контраст між текстом і фоном для легкого сприйняття інформації, зменшення навантаження на очі.
- Культурні особливості: значення кольорів може суттєво відрізнятися в різних країнах і культурах. Наприклад, білий колір асоціюється з чистотою на Заході, але може сприйматись як траурний в деяких азійських культурах.
- Аудиторія: потрібно орієнтуватися на очікування, смаки й уподобання цільових користувачів. Колірне рішення має відповідати їхнім емоційним реакціям.

Раціональний підхід до кольорового оформлення сайту сприяє створенню комфортного інтерфейсу, формує позитивне враження від ресурсу та сприяє досягненню бізнес-цілей [18].

Для підбору кольорів використовують так зване коло Йоганнеса Іттена, зображено на рисунку 3.1 [18]. Вибір кольорів відбувається за вершинами вписаних у коло трикутника або шестикутника. Обрані таким чином відтінки кольорів будуть пасувати одне одному.

Отже для розробки сайту було обрано таку кольорову гаму : чорний, блакитний, жовтий, оранжевий, сірий, коричневий та білий кольори. Оскільки розроблюваний сайт є інформаційним, то для користувача буде цікавим таке фонове зображення, яке буде відображати тематику сайту.

Для хорошої читабельності, при відкритті сторінки користувачем, контент буде відображатись сірим кольором на легко - прозорому булому фоні.



Рисунок 3.1 – Коло Йоганнеса Іттена

Підібрана колірна гамма, буде легкою для сприйняття відвідувача, оскільки основна увага буде акцентована на сірому тексті, який не буде засліплювати напівпрозорий білий фон.

Лістинг головної сторінки сайту надано на рисунках 3.2-3.3.

На рисунку 3.4 надано приклад головної сторінки WEB-ресурсу компанії із перевезення вантажів.

Розглянемо розробку динамічного контенту сайту.

Динамічний контент — це інформаційне наповнення веб-сторінки, яке може змінюватися в залежності від певних умов або дій користувача.

Початково веб-простір складався переважно зі статичних гіпертекстових документів, які не змінювались після завантаження. Проте з розвитком технологій веб-сторінки набули інтерактивних властивостей, що дозволило формувати їх вміст динамічно, відповідно до поточних сценаріїв взаємодії з користувачем.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

<head>
<title>Вантажні перевезення</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />
<link rel="stylesheet" type="text/css" href="style.css" />
</head>

<body>
<div class="width=100% height=100% align-left"></div><div
class="width=100% height=100% align-left"></div><div class="align-
left"></div>

    <div id="header">
        
        
        <ul>
            <li><a href="index.html"></a></li>
            <li><a href="page1.html"></a></li>
            <li><a href="page2.html"></a></li>
            <li><a href="page3.html"></a></li>
            <li><a href="page4.html"></a></li>
        </ul>
    </div>
    <div id="content">
        <div class="leftside">
            
            <div class="block border">

            </div>
            <div class="block">
                <br><br><br><br>
            </div>
        </div>
    </div>

```

Рисунок 3.2 - Лістинг головної сторінки сайту (початок)

```

<div class="rightside">

<div class="block pad">
<strong> <font size="3"> <h2>Вантажні
перевезення</h2> </font> </strong><br />
<p align="justify">

<font size="3">
<div id="abzac"> <p align="justify">Автомобільна
індустрія належить до основоположних галузей економіки країни. Від рівня
розвитку автомобільної промисловості наряду залежить рівень
економічного розвитку країни та благополуччя людей.
Спільне українське підприємство транспортно-
експедиційна компанія яка професійно та комплексно займається
перевезеннями вантажів а саме: негабаритних, великовагових та
нестандарних вантажів у міжнародному сполученні.</div>
<div id="abzac"> <p align="justify">На ринку
перевезення вантажів компанія працює з 2014 року. Компанія спеціалізується
на перевезенні вантажів з країн Європи до України.

Вантажні перевезення від компанії - це команда
професіоналів, яка допоможе організувати та надасть усі необхідні
консультації у процесі перевезення вантажів та дорожньої, будівельної,
сільськогосподарської техніки, промислового обладнання, яхт і човнів,
збірних та комплексних вантажів.</div>
<div id="abzac"> <p align="justify">Всі автомобілі
укомплектовані техстрапними ремнями для кріплення вантажу у
напівпричепи та причепи. Тентовані автомобілі обладнані комплектом АДР
для перевезень небезпечних вантажів. Все це гарантує своєчасну доставку
вантажів з найменшими втратами.
Усі водії підприємства пройшли навчання по
перевезенню небезпечних вантажів і мають візи на в'їзд до країн-учасниць
Шенгенського договору.
У визначенні власних пріоритетів ми завжди
орієнтуємось на потреби Клієнта та опираємось на багаторічний досвід у
сфері перевезення вантажів.
</div>
</font>
</p>
</div>
<br><br>
</div>

```

Рисунок 3.3 - Лістинг головної сторінки сайту (закінчення)

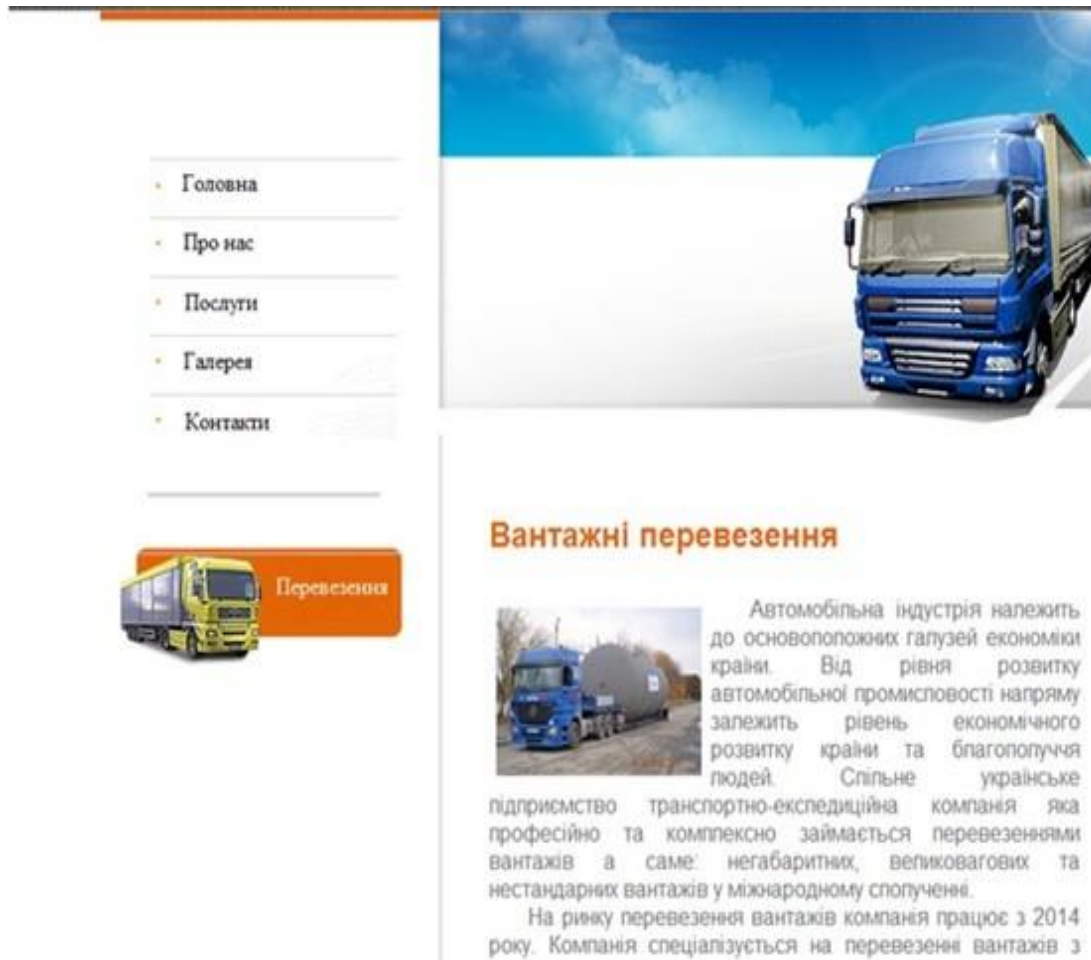


Рисунок 3.4 – Головна сторінка сайту

Існує два основні підходи до реалізації динамічного контенту.

1. Клієнтська динаміка: використання мов сценаріїв, що виконуються на стороні браузера (наприклад, JavaScript), дозволяє змінювати частини вмісту без необхідності перезавантаження всієї сторінки. Це забезпечує плавну і швидку взаємодію з користувачем.

2. Серверна динаміка: зміни на сторінці здійснюються за рахунок виконання програмного коду на сервері. Контент, який формується на основі даних, отриманих через форми, параметри URL, типи пристроїв, час доби тощо, генерується безпосередньо перед надсиланням сторінки в браузер користувача.

У першому випадку, як правило, використовуються технології, пов'язані з DHTML (Dynamic HTML), де за зміну елементів відповідають мови JavaScript або ActionScript. Для реалізації складніших мультимедійних ефектів раніше застосовували Flash.

У другому варіанті застосовуються серверні мови програмування, такі як PHP, ASP, JSP, Perl тощо. Ці мови забезпечують гнучке управління структурою сторінки ще до її відображення в браузері [17].

Запровадження динамічного контенту дозволяє підвищити інтерактивність веб-ресурсу, зробити його більш гнучким та адаптивним до потреб користувача.

Проте, надмірне використання динамічних елементів може негативно вплинути на продуктивність сайту, зокрема збільшити час завантаження сторінок та підвищити навантаження на клієнтські пристрої [17].

Для того щоб зацікавити відвідувача на одній з сторінок сайту створена галерея у вигляді таблиці в якій є зображення техніки нашої компанії, яка використовується для надання послуг вантажних перевезень (рис. 3.5). Зовнішній вигляд основних сторінок WEB-ресурсу компанії із перевезення вантажів «Про нас», «Послуги», «Галерея», «Контакти» надано на рисунках 3.7 – 3.9.

3.3 Тестування web-ресурсу

Останнім етапом розробки сайту є його тестування, що полягає у виявленні помилок, які не можна визначити на етапі розробки.

У процесі тестування веб-сайту, залежно від вимог технічного завдання, виконуються різноманітні перевірки.

Галерея нашого транспорту



Рисунок 3.5 – Реалізована галерея на сайті



Рисунок 3.6 – Зовнішній вигляд сторінки «Про нас»



Рисунок 3.7 – Зовнішній вигляд сторінки «Послуги»

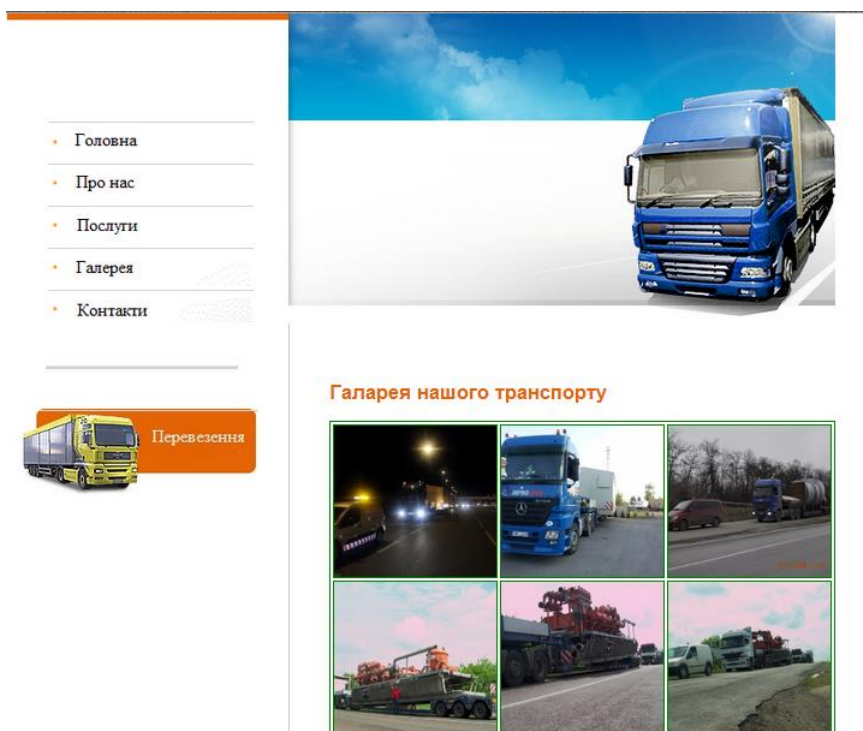


Рисунок 3.8 – Зовнішній вигляд сторінки «Галерея»



Рисунок 3.9 – Зовнішній вигляд сторінки «Контакти»

- Перевірка відображення шрифтів, анімації та графіки. Виявляються можливі проблеми з відтворенням текстових шрифтів, анімаційних елементів і зображень, які необхідно виправити для забезпечення стабільної роботи сайту.
- Аналіз відображення кольорів. Тестується коректність передачі кольорів на різних моніторах із різними налаштуваннями палітри, що дозволяє передбачити вигляд сайту у відвідувачів.
- Контроль змісту. Проводиться порівняння текстового та графічного наповнення сторінок із вихідними матеріалами замовника, а також перевірка орфографії та граматики.
- Перевірка метаданих. Аналізуються заголовки сторінок, ключові слова, описи та інші мета-теги для забезпечення їх відповідності SEO-вимогам.

- Тестування швидкості завантаження. Визначається час завантаження сторінок при різних швидкостях інтернет-з'єднання, що важливо для оптимізації користувацького досвіду.
- Оцінка відображення на різних моніторах. Аналізується зовнішній вигляд сайту на екранах з різними розмірами та роздільною здатністю, зокрема наявність горизонтальної прокрутки та коректне розташування елементів інтерфейсу.
- Перевірка гіперпосилань. Виявляються та усуваються «биті» або некоректні посилання, які ведуть на неправильні сторінки або не працюють.
- Тестування у різних браузерях. Перевіряється сумісність сайту з різними веб-браузерами та їхніми версіями, що дає змогу оцінити кросбраузерність та коректність відображення контенту.

На сьогоднішній день понад 95% користувачів працюють із такими веб-браузерами, як Internet Explorer, Google Chrome, Mozilla Firefox та Opera. Хоча ці програми зобов'язані дотримуватися встановлених стандартів для обробки HTML та CSS, іноді трапляється, що розробник пропускає певні деталі. У результаті цього браузери можуть або не відображати деякі властивості елементів, або показувати їх некоректно. Кросбраузерність є обов'язковою для кожного сайту.

Отже для початку впевнимось, що сайт буде однаково виглядати у різних браузерах, що є однією з вимог замовників [19]. У процесі тестування вебсайт коректно функціонував у всіх основних браузерах. Шрифти відображалися правильно, а зображення займали відповідні, передбачені для них місця на сторінці.

Для прикладу, зовнішній вигляд головної сторінки сайту в браузерах Google Chrome та Internet Explorer можна побачити на рисунках 3.10 та 3.11 відповідно.

Якщо провести порівняння прикладів наведених на рисунках 3.10 та 3.11, то можна зробити висновок, що вимога щодо однакового зовнішнього вигляду сайту в різних браузерах дотримана і це свідчить про хорошу якість розробленого веб-сайту. Також перевірено роботу переходів між сторінками сайту. Приклад результатів цього тестування зображено на рисунках: 3.5 - 3.6. Провівши тестування переходів між сторінками, можна зробити висновок, що все добре працює і користувач може з однієї сторінки перейти на будь-яку іншу, що свідчить про хорошу навігацію сайту.



Рисунок 3.10 – Зовнішній вигляд головної сторінки сайту у браузері Google Chrome

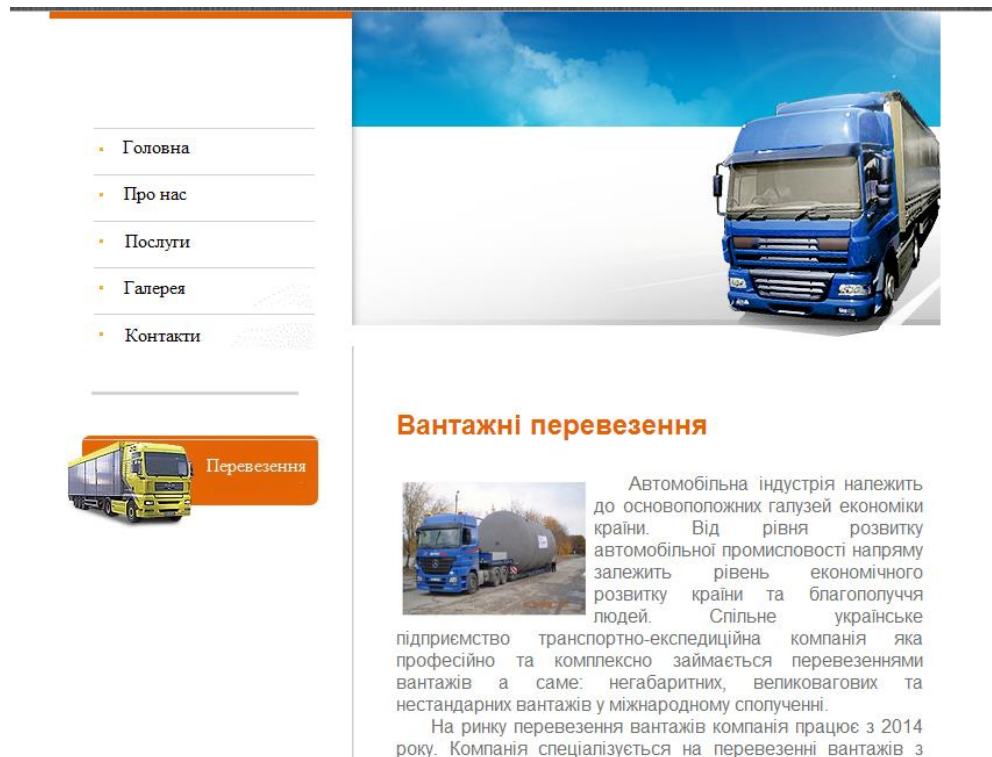


Рисунок 3.11 – Зовнішній вигляд головної сторінки сайту у браузері Internet Explorer

3.4 Висновок до розділу 3

Для оформлення сайту було обрано таку кольорову гаму: чорний, блакитний, жовтий, оранжевий, сірий, коричневий та білий кольори. Оскільки сайт є інформаційним, то для користувача буде цікавим таке фонове зображення, яке буде відображати тематику сайту. Для хорошої читабельності, при відкритті сторінки користувачем, контент буде відображатись сірим кольором на легко-прозорому булому фоні. Підібрана колірна гамма, буде легкою для сприйняття відвідувача, оскільки основна увага буде акцентована на сірому тексті, який не буде засліплювати напівпрозорий білий фон.

При розробці динамічного контенту обрано підхід, при якому відбувається використання скриптів, що виконуються в браузері користувача, для зміни вмісту сторінки в залежності від певних дій користувача. Для зміни

не потрібно повного перезавантаження сторінки. Обране рішення дозволяє використовувати скриптові мови для Dynamic HTML (DHTML) - JavaScript або ActionScript. Для додавання відео, звуків та графічних ефектів використано технологію Flash [11, 16, 17]. Для написання html-коду під час розробки було використано Notepad++, який надає переваги розробнику під час написання та редагування коду, а також легкий у використанні.

За допомогою Photoshop було розроблено логотип вебсайту, який чітко та ясно розкриває суть та відображає тематику розробленого вебсайту.

Після завершення розробки, було проведено успішне тестування на коректне відображення сайту та інформації, що він містить, у різних браузерах. Також проведено тестування переходу між сторінками. Проведено тестування переходів між сторінками та відображення сайту в різних браузерах. Результати тестування підтвердили працеспроможність та хорошу навігацію сайту.

ВИСНОВКИ

Поставлені перед бакалаврською кваліфікаційною роботою задачі виконано в повному обсязі [2], а саме:

- досліджено вже наявні аналогічні рішення;
- обрано та обґрунтовано підхід до вирішення поставленого завдання;
- розроблено структури та дизайн WEB-ресурсу компанії із перевезення вантажів;
- створено програму WEB-ресурсу компанії із перевезення вантажів;
- протестовано та перевірено функціонування ресурсу;
- створено користувацьку інструкцію з експлуатації WEB-ресурсу.

В ході виконання бакалаврської роботи були проаналізовані найпопулярніші системи для розборки WEB-ресурсу компанії із перевезення вантажів. Відповідно до завдання було проведено аналіз проблеми та обґрунтування необхідності даної розробки. Розроблено структуру та систему навігації сайту. Відповідно до усіх вимог було проведено розробку графічного забезпечення, що включало в себе розробку графіки та анімації а саме: текстур, логотипів, «шапки» сайту, різноманітних розділювачів та інших елементів дизайну. Було проаналізовано різні типи структур вебсайтів, обрано і розроблено оптимальну структуру вебресурсу та розроблено алгоритми функціонування вебресурсу. Відповідно з завданням до бакалаврської роботи був розроблений WEB-ресурс компанії із перевезення вантажів. Також було проведено тестування, яке підтвердило роботоспроможність WEB-ресурсу.

Мета роботи - розширення функціональних можливостей компанії WEB-ресурсу компанії із перевезення вантажів за рахунок реалізації дружнього та комфортного інтерфейса і збільшення галереї зображень досягнута.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. М.А. Гуменюк, В.О. Денисюк. Розробка сайту компанії перевізника вантажів. Молодь в науці: дослідження, проблеми, перспективи (МН-2025). Вінницький національний технічний університет. URL: <https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/viewFile/24622/20385> .
2. Методичні вказівки до виконання бакалаврських кваліфікаційних робіт для студентів спеціальності 122 «Комп'ютерні науки» [Електронний ресурс] / уклад.: А. А. Яровий, О. В. Сілагін, І. В. Богач. Вінниця: ВНТУ, 2023. (PDF, 54 с.). URL: https://iq.vntu.edu.ua/method/getfile.php?fname=124285.pdf&x=1&card_id=46522&id=124285 .
3. TVL. Вантажні перевезення. URL: <https://www.tvl.net.ua/en/nashi-poslugi/avtoperevezennya/>
4. STRAIGHT AUTO. Вантажні перевезення по Україні. URL: <https://straightauto.com.ua/en/services/ukraina.html>
5. 85 шаблонів і тем веб-сайтів для автоперевезень. URL: <https://www.templatemonster.com/ua/category/trucking-website-templates/?srsltid=AfmBOoo0tLQTly0I6NbVugJ3Idyncli5Y0DvacdDTBEtYWeNjl66JuSU>
6. Програмне забезпечення для управління вантажними тарифами: необхідний інструмент для сучасного бізнесу. URL: <https://www.cargoson.com/uk/blog/programne-zabezpechennya-dlya-upravlinnya-vantazhnimi-tarifami> .
7. Розробка сайту транспортної компанії. URL: <https://webstudio2u.net/ua/index/business/306-site-for-transport.html>
8. Основні етапи створення сайту. URL: <http://repair.lviv.ua/stvorenniya-sajtiv/osnovni-etapi-stvorenniya-sajtu/> .
9. Довідник по HTML тегам. URL: <https://css.in.ua/html/tags> .

10. Каскадні таблиці стилів. URL: http://uk.wikipedia.org/wiki/Каскадні_таблиці_стилів .
11. JavaScript. URL: <http://uk.wikipedia.org/wiki/JavaScript> .
12. Основні етапи створення сайту. URL: <http://repair.lviv.ua/stvorenniya-sajtiv/osnovni-etapi-stvorenniya-sajtu/> .
13. Веб-інтерфейс. URL: <http://uk.wikipedia.org/wiki/Веб-інтерфейс> .
14. Приклади дизайну веб-інтерфейсу. URL: <https://stfalcon.com/uk/blog/post/web-ui-design-examples> .
15. Веб студія WHITEWEB. URL: <https://www.whiteweb.ua/> .
16. Пасічник О.В., Пасічник В.В. Веб-дизайн: Підручник. - 2-ге вид., стер. Львів: ПП “Магнолія 2006”, 2024. 520 с.
17. Динамічна веб-сторінка. URL: http://uk.wikipedia.org/wiki/Динамічна_веб-сторінка .
18. Базові правила роботи з кольором для дизайну сайтів. URL: <https://skvot.io/uk/blog/glavnoe-o-tsvetovykh-skhemakh> .
19. Інструменти для тестування кросбраузерності верстки. URL: <https://training.qatestlab.com/blog/technical-articles/tools-for-testing-cross-browser-layout/> .

ДОДАТОК А (ОБОВ'ЯЗКОВИЙ)

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Розробка WEB-ресурсу компанії із перевезення вантажів

Тип роботи: бакалаврська кваліфікаційна робота

(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

Підрозділ кафедра комп'ютерних наук

(кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism 1,66 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

☒ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту

☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.

☐ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

Яровий А.А., зав. каф. КН

(прізвище, ініціали, посада)

Колесницький О.К., проф. каф КН

(прізвище, ініціали, посада)

(підпис)

(підпис)

Особа, відповідальна за перевірку

(підпис)

Озеранський В.С.

(прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

Керівник

(підпис)

Денисюк В.О.

(прізвище, ініціали, посада)

Здобувач

(підпис)

Гуменюк М.А.

(прізвище, ініціали)

ДОДАТОК Б (ОБОВ'ЯЗКОВИЙ)

ЛІСТИНГ ПРОГРАМИ

HTML-код головної сторінки (index.html)

```
<html lang="uk">
<head>
    <meta charset="windows-1251" />
    <title>Вантажні перевезення</title>
    <link rel="stylesheet" href="css/style.css" />
</head>
<body>

<div id="container">

    <div id="header">
        
        
        <ul id="nav">
            <li><a href="index.html"></a></li>
            <li><a href="page1.html"></a></li>
            <li><a href="page2.html"></a></li>
            <li><a href="page3.html"></a></li>
            <li><a href="page4.html"></a></li>
        </ul>
    </div>

    <div id="content">
        <div class="leftside">
```

```

        
        <div class="block border">
            <!-- Можна додати новини -->
        </div>
    </div>

    <div class="rightside">
        
        <div class="block pad">
            <h2>Вантажні перевезення</h2>
            
            <p>Автомобільна індустрія належить до
основоположних галузей економіки країни...</p>

            <p>На ринку перевезення вантажів
компанія працює з 2014 року...</p>

            
            <p>Всі автомобілі укомплектовані
техстропними ременями для кріплення вантажу...</p>
        </div>
    </div>
</div>

</body>
</html>

```

Стилі CSS (css/style.css):

```

body {
    font-family: Arial, sans-serif;
    background-color: #f8f8f8;
    margin: 0;
    padding: 0;
}

```

```
#container {
    width: 1000px;
    margin: 0 auto;
}

#header {
    background-color: #ffffff;
    padding: 10px;
}

#nav {
    list-style: none;
    padding: 0;
    margin-top: 10px;
}

#nav li {
    display: inline-block;
    margin-right: 5px;
}

.leftside, .rightside {
    float: left;
    padding: 10px;
}

.leftside {
    width: 30%;
}

.rightside {
    width: 65%;
}

.left-img {
    float: left;
    margin: 0 15px 10px 0;
}

.right-img {
    float: right;
    margin: 0 0 10px 15px;
}

h2 {
```

```

        color: #003366;
    }

    .block {
        background-color: #ffffff;
        padding: 15px;
        margin-top: 10px;
        border: 1px solid #ccc;
    }

    .clearfix::after {
        content: "";
        display: table;
        clear: both;
    }

```

`page1.html` — Про нас

```

<html lang="uk">
<head>
    <meta charset="windows-1251" />
    <title>Про компанію</title>
    <link rel="stylesheet" href="css/style.css" />
</head>
<body>

<div id="container">

    <div id="header">
        
        
        <ul id="nav">
            <li><a href="index.html"></a></li>
            <li><a href="page1.html"></a></li>

```

```

    <li><a href="page2.html"></a></li>
    <li><a href="page3.html"></a></li>
    <li><a href="page4.html"></a></li>
</ul>
</div>

```

```

<div id="content">
  <div class="rightside">
    <div class="block pad">
      <h2>Про компанію</h2>
      <p align="justify">
        Наша компанія надає послуги міжнародних вантажних
        перевезень з 2014 року.
        Ми спеціалізуємось на транспортуванні негабаритних,
        великовагових і небезпечних вантажів.
        Компанія має у власному розпорядженні сучасний
        автопарк та команду досвідчених водіїв.
      </p>
      <p align="justify">
        Основний принцип роботи – індивідуальний підхід до
        кожного клієнта,
        безпека вантажу та оперативність виконання замовлень.
      </p>
    </div>
  </div>
</div>

</div>

</body>
</html>

```

page2.html` — Послуги

```

<html lang="uk">
<head>
  <meta charset="windows-1251" />
  <title>Послуги</title>
  <link rel="stylesheet" href="css/style.css" />
</head>
<body>

<div id="container">

  <div id="header">
    <!-- той самий хедер -->
  </div>

  <div id="content">
    <div class="rightside">
      <div class="block pad">
        <h2>Наші послуги</h2>
        <ul>
          <li>Міжнародні вантажні перевезення (ЄС →
Україна)</li>
          <li>Перевезення негабаритних вантажів</li>
          <li>Транспортування сільськогосподарської
техніки</li>
          <li>Доставка будівельного обладнання</li>
          <li>Послуги експедирування</li>
          <li>Консультації та митне оформлення</li>
        </ul>
      </div>
    </div>
  </div>

</div>

</body>
</html>

```

`page3.html` — Автопарк

```

<html lang="uk">
<head>
  <meta charset="windows-1251" />
  <title>Автопарк</title>
  <link rel="stylesheet" href="css/style.css" />
</head>
<body>

<div id="container">

  <div id="header">
    <!-- той самий хедер -->
  </div>

  <div id="content">
    <div class="rightside">
      <div class="block pad">
        <h2>Наш автопарк</h2>
        <p>У розпорядженні компанії:</p>
        <ul>
          <li>Тентовані вантажівки (ADR-комплектація)</li>
          <li>Низькорамні трали</li>
          <li>Контейнеровози</li>
          <li>Платформи для негабаритного транспорту</li>
        </ul>
        <p>
          Весь транспорт проходить регулярне
техобслуговування та відповідає європейським стандартам безпеки.
        </p>
      </div>
    </div>
  </div>

</div>

```

```
</body>
</html>
```

`page4.html` — Контакти

```
<html lang="uk">
<head>
  <meta charset="windows-1251" />
  <title>Контакти</title>
  <link rel="stylesheet" href="css/style.css" />
</head>
<body>

<div id="container">

  <div id="header">
    <!-- той самий хедер -->
  </div>

  <div id="content">
    <div class="rightside">
      <div class="block pad">
        <h2>Контактна інформація</h2>
        <p><strong>Адреса:</strong> м. Київ, вул. Транспортна,
12</p>
        <p><strong>Телефон:</strong> +38 (044) 123-45-67</p>
        <p><strong>Email:</strong> info@transport.com.ua</p>
        <p><strong>Графік роботи:</strong> Пн-Пт, 9:00–
18:00</p>

        <h3>Форма зворотного зв'язку</h3>
        <form action="#" method="post">
          <p><input type="text" name="name" placeholder="Ваше
ім'я" required></p>
          <p><input type="email" name="email" placeholder="Ваш
email" required></p>
```

```
        <p><textarea name="message" rows="5"
placeholder="Ваше повідомлення"></textarea></p>
        <p><input type="submit" value="Надіслати"></p>
    </form>
</div>
</div>
</div>

</div>

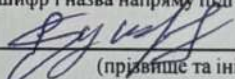
</body>
</html>
```

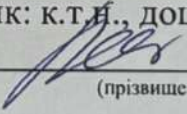
ДОДАТОК В (ОБОВ'ЯЗКОВИЙ)

ІЛЮСТРАТИВНА ЧАСТИНА

РОЗРОБКА WEB-РЕСУРСУ КОМПАНІЇ
ІЗ ПЕРЕВЕЗЕННЯ ВАНТАЖІВ

Виконав: студент 4-го курсу,
групи ЗКН-22м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напрямку підготовки, спеціальності)

 Гуменюк М.А.
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН
 Денисюк В.О.
(прізвище та ініціали)

« 03 » 06 2025 р.

Вінниця ВНТУ - 2025 рік



Рисунок В.1 – Розширена організаційна структура компанії

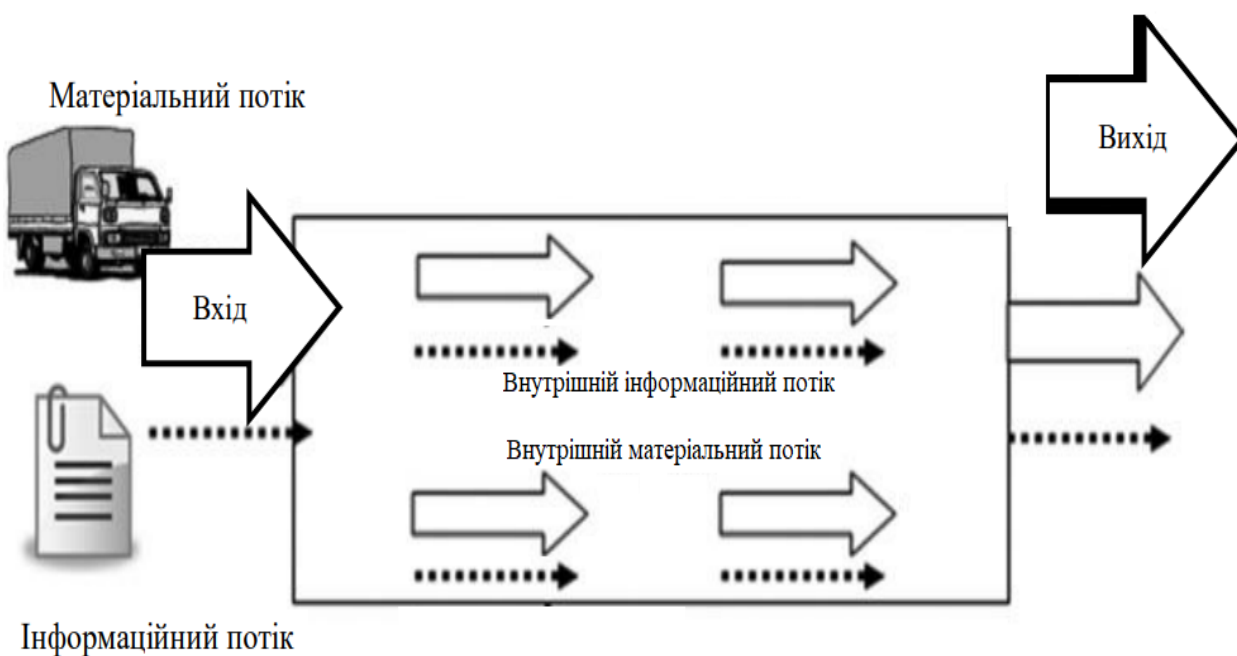


Рисунок В.2 – Потокова схема транспортної системи

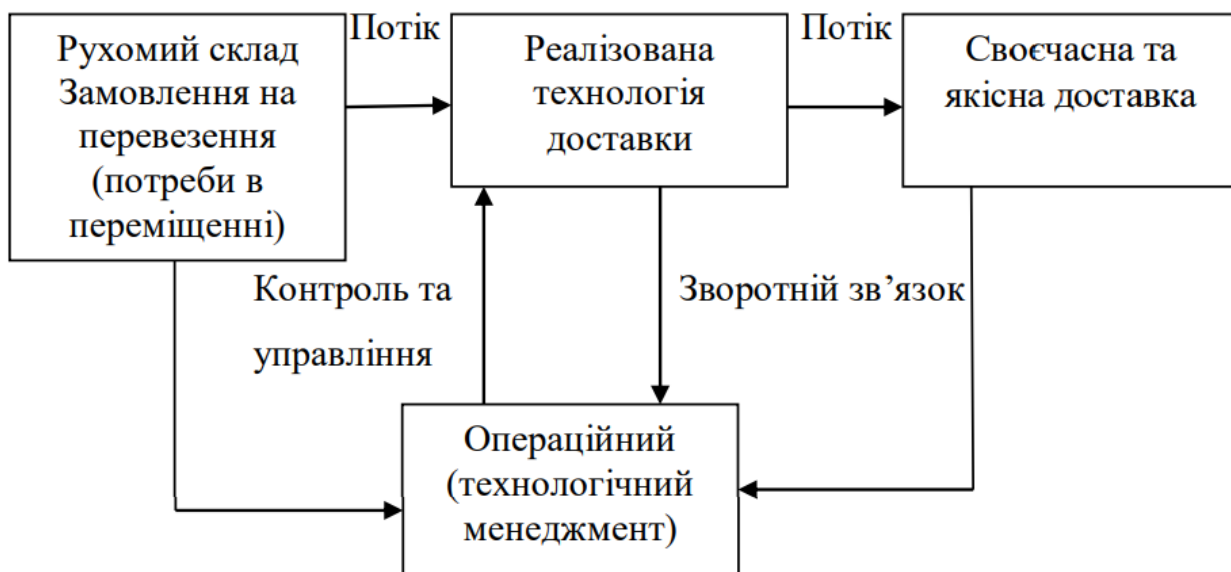


Рисунок В.3 - Операційна схема доставки товарів

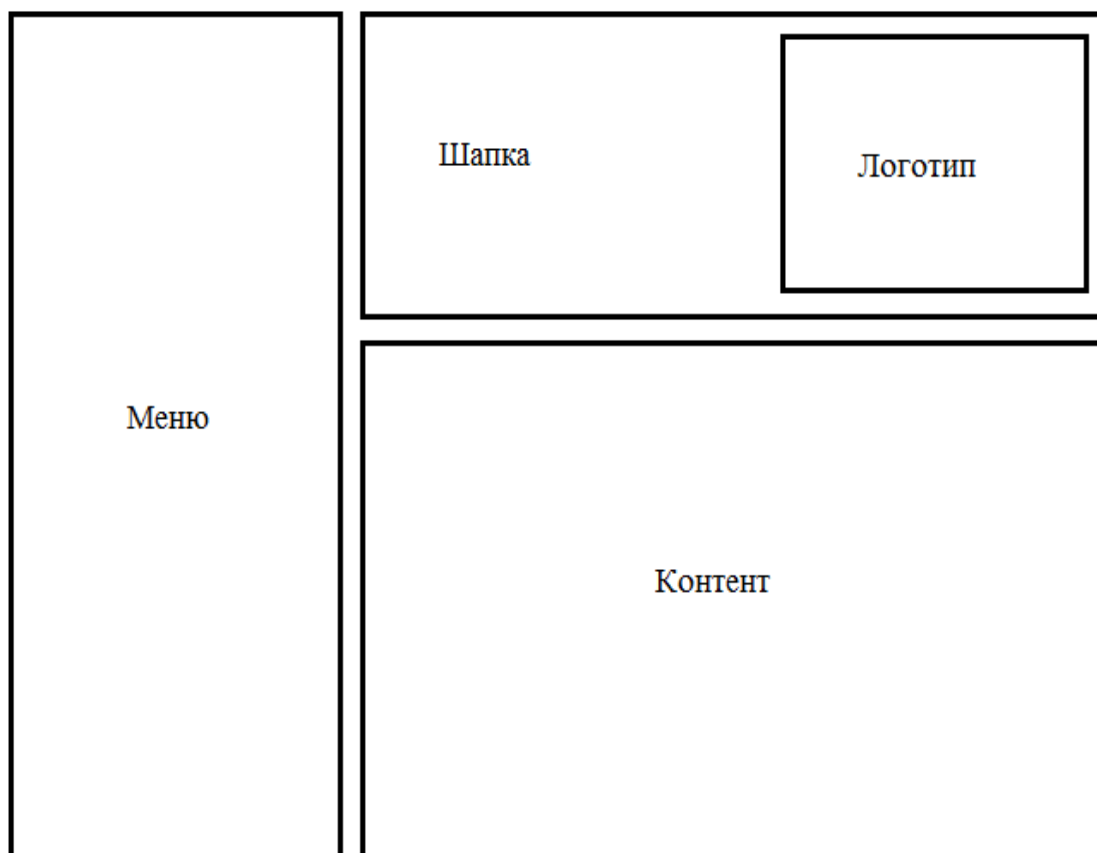


Рисунок В.4 – Вигляд сторінки сайту

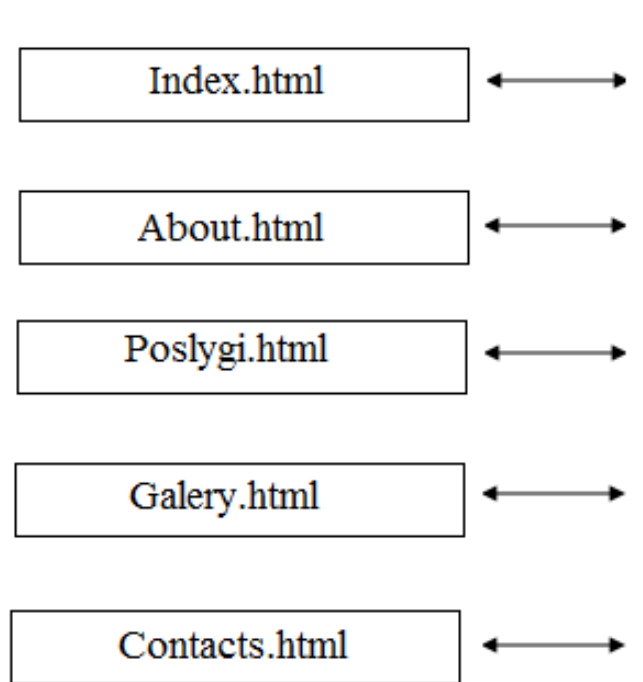


Рисунок В.5 – Схема структуры веб-сайту

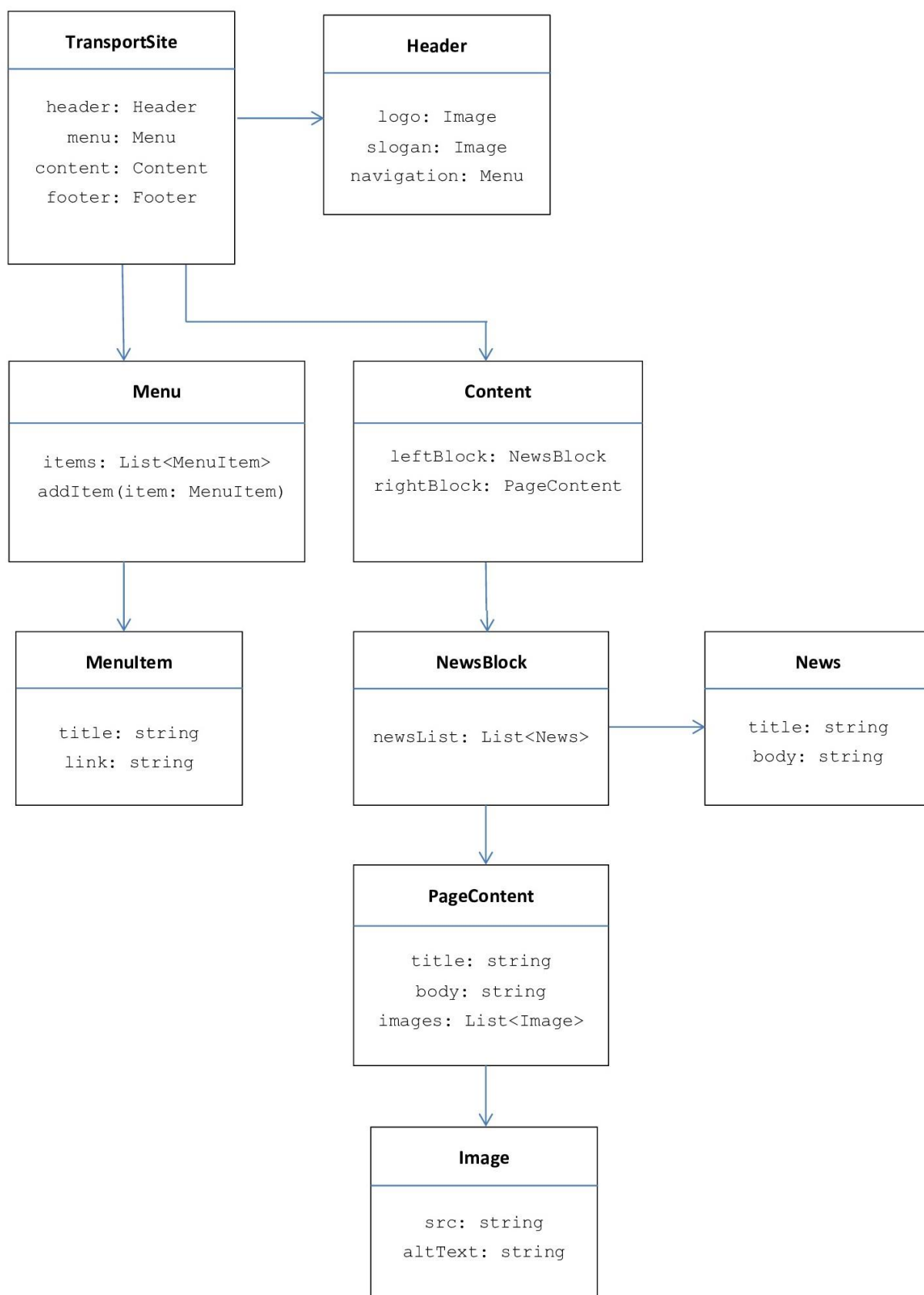


Рисунок В.6 - Структура класів для сайту "Вантажні перевезення"

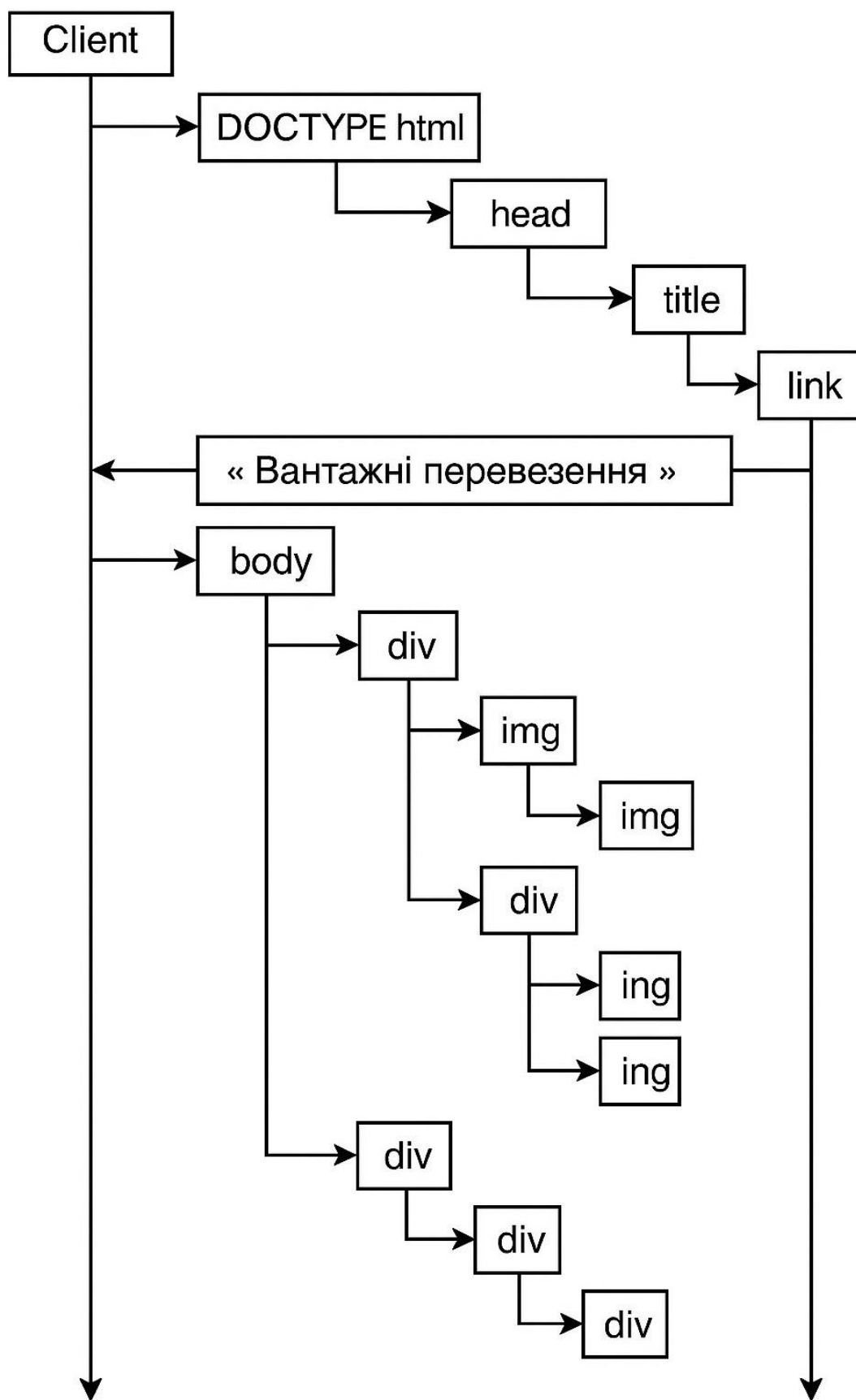


Рисунок В.7 - UML-діаграма послідовності дій



Рисунок В.8 – Коло Йоганнеса Іттена



Рисунок В.9 – Головна сторінка сайту

Галерея нашого транспорту



Рисунок В.10 – Реалізована галерея на сайті



Рисунок В.11 – Зовнішній вигляд сторінки «Про нас»



Рисунок В.12 – Зовнішній вигляд сторінки «Послуги»

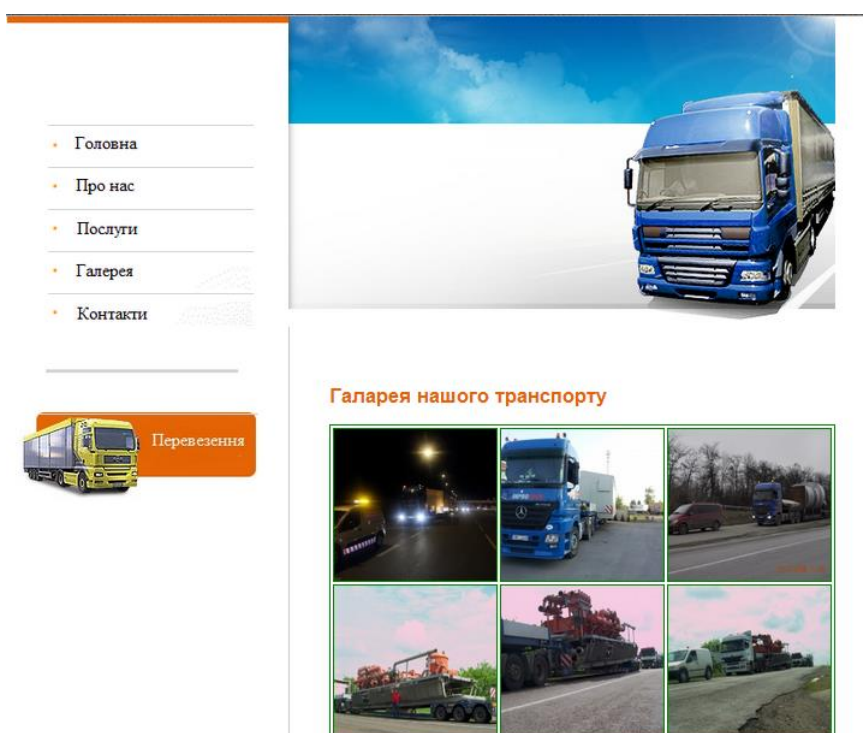


Рисунок В.13 – Зовнішній вигляд сторінки «Галерея»



Рисунок В.14 – Зовнішній вигляд сторінки «Контакти»

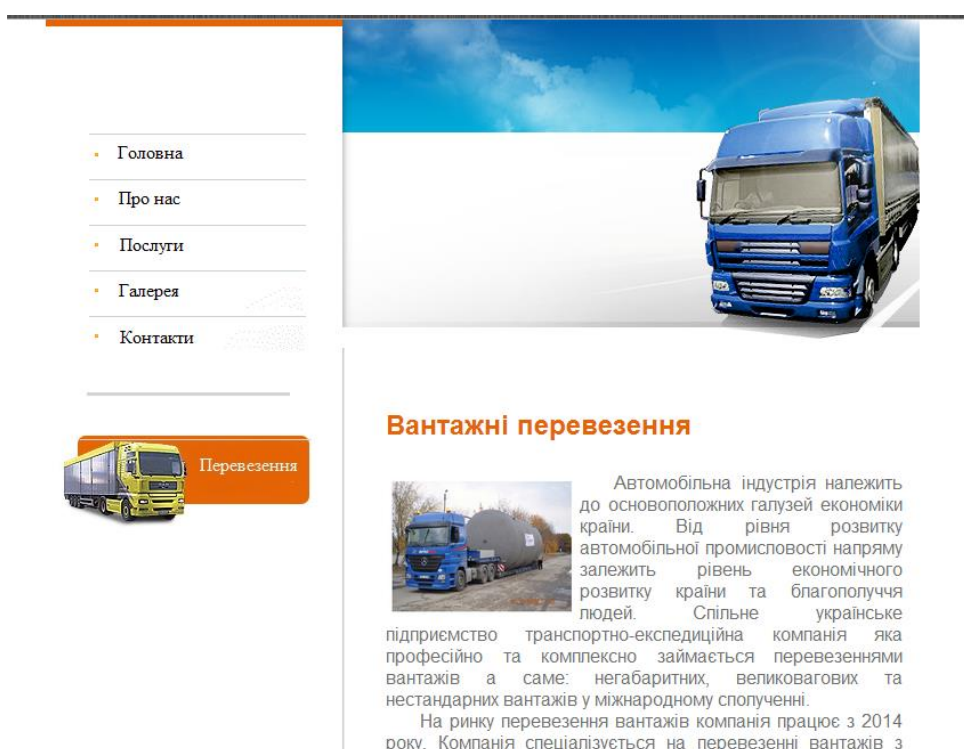


Рисунок В.15 – Зовнішній вигляд головної сторінки сайту у браузері Google Chrome та Internet Explorer при тестуванні.

ДОДАТОК Г (ДОВІДНИКОВИЙ)

ІНСТРУКЦІЯ КОРИСТУВАЧА

Оформлення заявки на вантажне перевезення відбувається за таким сценарієм.

Основні кроки такі.

1. Відкриття сайту.
2. Перехід у розділ "Вантажні перевезення".
3. Ознайомлення з інформацією про послуги.
4. Натискання кнопки "Замовити перевезення".
5. Заповнення форми (дані вантажу, маршрут, контакти).
6. Надсилання заявки.
7. Підтвердження успішного відправлення (відповідь системи).

Інструкція користувача, відповідно, містить такі кроки:

Початок

- "Відкрити сайт"
- "Перейти до розділу Вантажні перевезення"
- "Переглянути опис послуги"
- "Натиснути кнопку 'Замовити'"
- "Заповнити форму"
- "Надіслати заявку"
- "Показати повідомлення про успіх"

Кінець.