

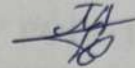
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук

Бакалаврська кваліфікаційна робота на тему:

РОЗРОБКА WEB-РЕСУРСУ СЛУЖБИ ПРОКАТУ АВТОМОБІЛІВ

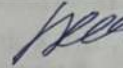
Виконав: студент (4) курсу, групи ЗКН-216
спеціальності 122 – Комп'ютерні науки

(шифр і назва напрямку підготовки, спеціальності)



Подольан А.Ю.
(прізвище та ініціали)

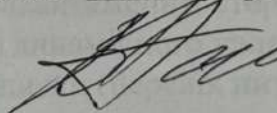
Керівник: к.т.н., доцент каф.КН



Денисюк В.О.
(прізвище та ініціали)

« 4 » червня 2025 р.

Рецензент: к.т.н., професор каф. АІІТ



Папінов В.М.
(прізвище та ініціали)

« 5 » червня 2025 р.

Допущено до захисту

Зав. кафедри

« 6 » 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

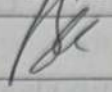
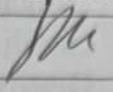
ЗАТВЕРДЖУЮ

Завідувач кафедри КН
проф., д.т.н. Яровий А.А.
«05» травня 2025 р.

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Подольану Антону Юрійовичу

1. Тема роботи: Розробка WEB-ресурсу служби прокату автомобілів.
Керівник роботи: Денисюк Валерій Олександрович, к.т.н., доцент
затверджені наказом вищого навчального закладу “20” березня 2025 року №97
2. Строк подання студентом роботи “30” травня 2025 р.
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 «05» травня 2025 року

КАЛЕНДАРНИЙ ПЛАН

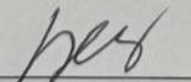
№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз предметної області WEB-ресурсу служби прокату автомобілів	05.05.25	07.05.25	
2	Постановка задачі	08.05.25	10.05.25	
3	Розробка програмного модуля WEB-ресурсу служби прокату автомобілів	10.05.25	15.05.25	
4	Обґрунтування засобів розробки WEB-ресурсу служби прокату автомобілів	16.05.25	18.05.25	
5	Програмна реалізація WEB-ресурсу служби прокату автомобілів	19.05.25	24.05.25	
6	Тестування WEB-ресурсу служби прокату автомобілів	25.05.25	28.05.25	
7	Розробка інструкції користувача	29.05.25	31.05.25	
8	Оформлення матеріалів до захисту БКР	01.06.25	04.06.25	

Студент


(підпис)

Подолян А.Ю.
(прізвище та ініціали)

Керівник роботи


(підпис)

Денисюк В.О.
(прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська кваліфікаційна робота складається з 92 сторінок формату А4, на яких є 15 рисунків, список використаних джерел містить 40 найменувань.

У роботі розглянуто актуальну задачу розробки WEB-ресурсу служби прокату автомобілів. На розробленому сайті представлена статична інформація, яка відповідає тематиці сайту, та динамічна інформація, яка розкриває тему сайту. Для створення та обробки графічних зображень було використано Adobe Photoshop CS5, HTML, CSS, JavaScript, редактор Notepad++.

Ключові слова: веб-додаток, прокат автомобілів, HTML, CSS, Java, Photoshop.

ABSTRACT

The bachelor's qualification work consists of 92 pages of A4 format, on which there are 15 figures, the list of used sources contains 40 names.

The paper considers the current task of developing a WEB resource for car rental services. The developed site presents static information that corresponds to the site's theme, and dynamic information that reveals the site's theme. Adobe Photoshop CS5, HTML, CSS, JavaScript, and Notepad++ editor were used to create and process graphic images.

Keywords: web application, car rental, HTML, CSS, Java, Photoshop.

ЗМІСТ

ВСТУП.....	3
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ПРОКАТУ АВТОМОБІЛІВ.....	5
1.1 Аналіз галузі WEB-ресурсів.....	5
1.2 Аналіз відомих програм аналогів.....	14
1.3 Постановка задачі.....	21
1.4 Висновок до розділу 1.....	23
2 РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ WEB-РЕСУРСУ	
СЛУЖБИ ПРОКАТУ АВТОМОБІЛІВ.....	24
2.1 Розробка структури WEB-ресурсу.....	24
2.2 Розробка діаграми класів WEB-ресурсу	
служби прокату автомобілів.....	29
2.3 Послідовність дій WEB-ресурсу	
служби прокату автомобілів.....	34
2.4 Висновок до розділу 2.....	36
3 ПРОГРАМНА РЕАЛІЗАЦІЯ WEB-РЕСУРСУ	
СЛУЖБИ ПРОКАТУ АВТОМОБІЛІВ.....	37
3.1 Обґрунтування вибору мови програмування.....	37
3.2 Створення WEB-ресурсу.....	47
3.3 Тестування WEB-ресурсу.....	59
3.4 Висновок до розділу 3	63
ВИСНОВКИ.....	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	65
ДОДАТОК А (ОБОВ'ЯЗКОВИЙ) ПРОТОКОЛ ПЕРЕВІРКИ	
КВАЛІФІКАЦІЙНОЇ РОБОТИ	68
ДОДАТОК Б (ОБОВ'ЯЗКОВИЙ) ЛІСТИНГ ПРОГРАМИ.....	69
ДОДАТОК В (ОБОВ'ЯЗКОВИЙ) ІЛЮСТРАТИВНА ЧАСТИНА.....	81
ДОДАТОК Г (ДОВІДНИКОВИЙ) ІНСТРУКЦІЯ КОРИСТУВАЧА.....	90

ВСТУП

Актуальність досліджень. На сьогодні Internet відіграє роль засобів масової інформації. В деяких аспектах поширення інформації через Internet є більш продуктивним ніж газети та журнали. Користуючись інтернетом можна шукати саме ту інформацію, яка цікавить користувача, в інтернеті новини з'являються на багато швидше ніж в газетах, через інтернет можна обговорювати новини з іншими користувачами, ділитися інформацією. Послуги з прокату автомобілів стали незамінними в різних секторах, таких як туризм, бізнес, транспорт, логістика. Оскільки все більше користувачів шукають гнучкі варіанти подорожей, продумано розроблений веб-сайт може спростити процес прокату, дозволяючи клієнтам легко знаходити та бронювати автомобілі, які відповідають їхнім потребам. Незалежно від того, чи вже є існуючий бізнес з прокату автомобілів, чи він тільки починає, безперебійний процес бронювання може підвищити задоволеність клієнтів та стимулювати розвиток бізнесу. Сьогодні більшість із нас звертається до Інтернету практично для всього, включаючи онлайн-бронювання послуг. Саме тому онлайн-системи прокату автомобілів стали такими популярними. Незалежно від того, чи потрібен людям автомобіль для ділової поїздки чи відпустки, вони люблять свободу та гнучкість, які надає оренда транспортного засобу. Отже, з вищенаведеного можна зробити висновок, що розробка WEB-ресурсу служби прокату автомобілів є актуальною та потребує подальшого дослідження.

Метою дослідження є розширення функціональних можливостей WEB-ресурсу служби прокату автомобілів за рахунок розширення каталогу автомобілів і використання дружнього та комфортного інтерфейсу.

Об'єктом дослідження є процеси, які виконуються WEB-ресурсом служби прокату автомобілів.

Предметом дослідження є алгоритми та програмне забезпечення, що

втілює WEB-ресурс служби прокату автомобілів.

Задачі дослідження:

- дослідження вже наявних аналогічних рішень;
- вибір та обґрунтування підходу до вирішення поставленого завдання;
- розробка структури та дизайну WEB-ресурсу служби прокату автомобілів;
- програмне створення WEB-ресурсу служби прокату автомобілів;
- 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 WEB-ресурсу служби прокату автомобілів» [1].

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

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ПРОКАТУ АВТОМОБІЛІВ

1.1 Аналіз галузі WEB-ресурсів

Сайт для оренди автомобілів — це інтернет-ресурс, що надає користувачам можливість забронювати автотранспорт на визначений час [3, 4]. Такі платформи зазвичай включають функціонал з переглядом характеристик авто, системою онлайн-бронювання і оплати, даними про пункти видачі/повернення машин, а також сервісом підтримки користувачів. Основною метою цього ресурсу є забезпечення зручного, швидкого та ефективного процесу оренди як для клієнтів, так і для компаній, що надають такі послуги.

Головні кроки для створення сайту прокату автомобілів такі:

- обрати доменне ім'я та хостинг;
- обрати конструктор веб-сайтів та шаблон (самостійна розробка);
- створити свій веб-сайт прокату автомобілів;
- створити привабливу головну сторінку;
- додати перелік інвентарю та деталей транспортних засобів;
- додати функцію бронювання та оплати;
- створити сторінку контактів;
- виконати оптимізацію для локального SEO (Search engine optimization);
- постійно просувати вашого веб-сайту з прокату автомобілів.

Розглянемо кроки створення сайту прокату автомобілів детально.

Обрати доменне ім'я та хостинг.

Міцна основа для веб-сайту з прокату автомобілів починається з доменного імені та веб-хостингу. Доменне ім'я служить онлайн-адресою, допомагаючи клієнтам знаходити вас в Інтернеті. Треба обрати домен, який

легко запам'ятовується, представляє бізнес і містить ключові слова, такі як «прокат автомобілів», щоб підвищити рейтинг у пошукових системах. Використовуйте пошук доменних імен , щоб переконатися, що вибраний вами домен є унікальним і релевантним, включаючи домен верхнього рівня (наприклад, .com), який ви, можливо, захочете використовувати. Генератор доменних імен може допомогти в мозковому штурмі креативних та релевантних імен.

Веб-хостинг - це місце, де зберігаються файли веб-сайту та вони доступні відвідувачам, тому забезпечення надійності сайту є критично важливим. Виберіть хостинг-сайт, який пропонує високий рівень безвідмовної роботи, відмінну підтримку клієнтів та можливості масштабування . З правильним налаштуванням домену та хостингу ви будете готові створити веб-сайт, який буде доступним та надійним для клієнтів сайту.

Обрання конструктор веб-сайтів та шаблон (власна розробка).

Створення професійно оформленого сайту стало доступнішим, ніж будь-коли, завдяки різноманітним конструкторам веб-сайтів. Ці платформи пропонують інтерфейси перетягування, готові шаблони та налаштовуваний дизайн, що дозволяє вам створювати свій сайт без зусиль. Вибираючи платформу, враховуйте такі аспекти, як вартість, простота використання та доступні функції. Використовуючи Wix, ви можете зручно керувати своїм бізнесом з будь-якого місця з Wi-Fi, як на комп'ютері, так і через мобільний додаток Wix .

Після того, як ви обрали конструктор веб-сайтів , виберіть шаблон веб-сайту , який відповідає вашому бізнесу з прокату автомобілів та бренду. Ви можете налаштувати шрифти, зображення та макети веб-сайту відповідно до вашого стилю. Така персоналізація гарантує, що ваш сайт демонструватиме унікальну ідентичність вашого бренду, водночас надаючи професійний та вишуканий вигляд вашим відвідувачам. макети веб-сайтів, що відповідають

вашому стилю. Така персоналізація гарантує, що ваш сайт демонструватиме виразну ідентичність вашого бренду, водночас надаючи відвідувачам професійний та вишуканий вигляд.

Розробка дизайн веб-сайту для прокату автомобілів.

Персоналізація шаблону може оживити ваш бренд прокату автомобілів, інтегрувавши ваш логотип, колірну схему та інші візуальні елементи у веб-дизайн. Переконайтеся, що дизайн залишається чистим та лаконічним, щоб відвідувачам було легко орієнтуватися та знаходити те, що вони шукають.

Щоб відобразити унікальну ідентичність вашого бренду та персоналізувати дизайн вашого веб-сайту з прокату автомобілів прокату автомобілів, розгляньте такі кроки.

- 1) Додати власну кольорову палітру. Узгодити кольорову палітру веб-сайту з брендом прокату автомобілів. Уникти відволікаючих або яскравих кольорів; натомість обрати заспокійливі та теплі тони, щоб покращити дизайн користувацького досвіду .
- 2) Обрати відповідні шрифти, типографіку, яку легко читати та яка відповідає бренду. Для професійного вигляду обрати чистий та класичний шрифт. Якщо сервіс прокату автомобілів має веселу та креативну атмосферу, розглянути унікальний та сучасний шрифт. Також можливо розглянути курсивні шрифти для більш елегантного вигляду або шрифти для плакатів, щоб створювати сміливі, привабливі заяви.
- 3) Спростити навігацію сайтом. Створити веб-сайт прокату автомобілів відповідно до процесу бронювання запитів клієнтів, спрощуючи їхній пошук інформації. Переконайтеся, що панель навігації легкодоступна, і використовуйте випадаючі меню для підкатегорій, щоб спростити навігацію сайтом.
- 4) Розмістити свій логотип на видному місці. Власний логотип представляє бізнес з прокату автомобілів, тому він має бути

професійним та незабутнім. Використовувати високоякісне зображення та розмістити його на видному місці на веб-сайті вашого прокату автомобілів, наприклад, у заголовку або нижньому колонтитулі.

Створити привабливу головну сторінку.

Власна домашня сторінка слугує фронтальною сторінкою інтернет-магазину, запрошуючи відвідувачів ознайомитися з веб-сайтом компанії та дізнатися більше про послуги з прокату автомобілів. Під час створення головної сторінки враховуйте такі елементи:

Додати високоякісні зображення: використовувати привабливі фотографії, які виділяють автомобілі компанії, послуги та місця призначення. Розглянути можливість використання засобу зміни розміру зображень для забезпечення оптимального відображення.

Інформувати про поточні акції або розпродажі, заохочувати клієнтів до дії, показуючи спеціальні пропозиції або пропозиції обмеженої за часом на головній сторінці сайту.

Спростити бронювання, додати помітну кнопку із закликом до дії, таку як «Забронювати зараз» або «Забронювати автомобіль», яка спрямовуватиме відвідувачів на веб-сторінку бронювання.

Зосередитися на відгуках клієнтів, позитивні відгуки інших користувачів зміцнюють довіру та авторитет, впливаючи на рішення потенційних клієнтів та сприяючи відчуттю надійності послуг.

Виконувати підтримку у чаті, покращувати обслуговування клієнтів і потенційно збільшувати продажі, пропонуючи підтримку в режимі реального часу за допомогою функції чату. Це дозволяє відвідувачам отримувати миттєві відповіді на свої запитання та проблеми, сприяючи довірі та залученості.

Виконувати відображення карти зони обслуговування. Якщо компанія з прокату автомобілів має кілька філій, відображення карти на головній

сторінці допоможе клієнтам легко знайти найближче до них місцезнаходження.

Надавати перелік транспортних засобів та їх опис.

Забезпечувати потенційним клієнтам доступ до всієї необхідної інформації про ваші послуги з прокату автомобілів, створивши детальну сторінку з описом автомобілів. Ця сторінка надає користувачам можливість порівняти різні характеристики автомобіля, перш ніж забронювати його в оренду. Включіть такі деталі, як марка та модель автомобіля, рік випуску, ціна за день або тиждень та будь-які спеціальні функції (наприклад, GPS, дитяче крісло). Додати фільтри для зручного пошуку, такі як ціновий діапазон, тип автомобіля, місткість та паливна економічність, щоб покращити взаємодію з користувачем .

Додати функцію бронювання та оплати.

Веб-сайт з прокату автомобілів повинен дозволяти клієнтам завершити весь процес онлайн-бронювання, від вибору автомобіля до здійснення оплати. Ця зручність може призвести до збільшення кількості бронювань та підвищити ймовірність задоволення клієнтів.

Щоб створити безперебійний процес бронювання на веб-сайті прокату автомобілів необхідно виконати такі вимоги:

- створити зручну для користувача форму бронювання, розробити прості онлайн-форми, які збирають важливу інформацію, таку як місце посадки та посадки, дата та час, а також контактні дані;
- створити пакети та плани, запропонувати різні пакети оренди – такі як щоденні, тижневі або щомісячні, або окремі плани для поїздки в один бік, місцевих та міжміських поїздки; врахувати різні потреби та вподобання клієнтів;
- включити варіанти оплати, дозволити клієнтам оплачувати бронювання безпосередньо на нашому веб-сайті прокату

автомобілів, використовуючи онлайн-методи обробки платежів, такі як кредитна картка, PayPal або Stripe.

Створити сторінку контактів та форми для потенційних клієнтів.

Спеціальна сторінка контактів є важливим елементом вашого веб-сайту з прокату автомобілів, оскільки вона дозволяє клієнтам звертатися з будь-якими питаннями чи занепокоєннями. Необхідно вказати свої місцезнаходження, номер робочого телефону, робочу електронну пошту та профілі в соціальних мережах. Використовувати форми збору потенційних клієнтів та CRM, таку як Wix, для ефективного управління взаємодією з клієнтами.

Ось кілька вимог щодо ефективного залучення потенційних клієнтів у бізнесі з прокату автомобілів:

- зробити все просто, створити форму лаконічною, запитуючи лише основні дані, такі як ім'я, електронна пошта, номер телефону та бажаний транспортний засіб;
- використовувати переконливий заклик до дії, використовувати привабливі заклики до дії, такі як «Дізнатися більше» або «Отримати цінову пропозицію», щоб мотивувати потенційних клієнтів заповнити форму;
- стимулювати реєстрації, пропонувати стимул, наприклад, знижку чи додаткову послугу, щоб заохотити більше людей заповнити форму та збільшити конверсії;
- впроваджувати ефективне подальше спостереження, впроваджувати систему для швидкого та ефективного подальшого зв'язку з потенційними клієнтами за допомогою телефонних дзвінків, електронних листів або інших методів зв'язку.

Оптимізація для локального SEO.

Оптимізація нашого веб-сайту з прокату автомобілів для локального SEO має вирішальне значення для залучення клієнтів у вашій зоні обслуговування. Пошукова оптимізація (SEO - Search engine optimization) допомагає вам займати вищі позиції на сторінках результатів пошукової видачі, що полегшує потенційним клієнтам пошук і бронювання ваших послуг.

Вимоги щодо покращення локального SEO.

1) Використовувати ключові слова, що стосуються певного місцезнаходження: додати ключові слова, що визначають райони, де надаються послуги, наприклад, «прокат автомобілів у Вінниці» або «оренда автомобіля поблизу аеропорту Бориспіль».

2) Заявити на отримання інформації про свою компанію в Google Мій бізнес: цей безкоштовний інструмент дозволяє керувати тим, як наша компанія відображається на сторінці результатів пошуку Google, включаючи відгуки, фотографії та контактну інформацію.

3) Створювати високоякісний контент: пошукові системи віддають перевагу веб-сайтам із цінним та інформативним контентом. Завести блог або створити корисні ресурси, пов'язані з орендою автомобілів та подорожами.

4) Забезпечити зручність для мобільних пристроїв: зі зростанням кількості людей, які використовують свої телефони для бронювання, вкрай важливо мати веб-сайт, зручний для мобільних пристроїв, який добре виглядає та функціонує на всіх пристроях.

5) Заохочувати відгуки клієнтів: позитивні відгуки не лише зміцнюють довіру потенційних клієнтів, але й покращують ваші рейтинги в пошукових системах. Заохочувати задоволених клієнтів залишати відгуки на нашому веб-сайті та інших сайтах відгуків, таких як Google My Business, Yelp та TripAdvisor тощо.

Постійно просувати веб-сайт з прокату автомобілів.

Просувати свій веб-сайт з прокату автомобілів за допомогою різних маркетингових стратегій, щоб охопити ширшу аудиторію та збільшити трафік на наш веб-сайт, що зрештою призведе до збільшення кількості потенційних клієнтів та конверсій для вашого бізнесу з прокату автомобілів.

Щоб максимізувати охоплення та видимість необхідно:

- використовувати соціальні мережі, такі платформи, як Facebook, Instagram та Twitter, щоб демонструвати свої автомобілі, ділитися відгуками клієнтів та акціями, а також взаємодіяти з потенційними клієнтами;
- проводити маркетингові кампанії електронною поштою, інформувати існуючих клієнтів про спеціальні пропозиції або майбутні події, надсилаючи регулярні розсилки;
- інвестувати в PPC-рекламу, розглянути можливість використання платної реклами в Google або соціальних мережах, щоб охопити ширшу аудиторію та орієнтуватися на певні демографічні групи;
- співпрацювати з місцевим бізнесом, з готелями, туристичними компаніями або іншими підприємствами, які орієнтовані на схожі демографічні групи клієнтів, для взаємного просування послуг.

Бізнесу з прокату автомобілів потрібен вебсайт.

Добре розроблений та оптимізований веб-сайт може вивести бізнес з прокату автомобілів на новий рівень, забезпечуючи цінний актив для залучення та утримання клієнтів.

Ось головні причини, чому бізнесу з прокату автомобілів потрібен веб-сайт.

1) Розширює власну клієнтську базу: завдяки сильній онлайн-присутності бізнес з прокату автомобілів може знайти потенційних клієнтів, які знаходяться далеко за межами нашого безпосереднього географічного регіону. Це означає, що можна залучати клієнтів з різних міст, штатів або навіть країн, значно розширюючи свою клієнтську базу.

2) Підвищення довіри: інвестування в добре розроблений веб-сайт з прокату автомобілів має вирішальне значення для підвищення довіри. Розміщуючи відгуки та рекомендації клієнтів на видному місці на вашому сайті, ви можете вселити довіру потенційним клієнтам. Ці позитивні рекомендації можуть запевнити нових клієнтів у якості вашого сервісу, що зрештою підвищить коефіцієнт конверсії.

3) Підвищення зручності для клієнтів: пропонування варіантів онлайн-бронювання та оплати значно підвищує зручність для ваших клієнтів. Це дозволяє їм бронювати будь-коли та з будь-якого місця, без необхідності особисто відвідувати чи телефонувати у ваш заклад. Це спрощує процес бронювання, що підвищує ймовірність того, що клієнти оберуть саме ваш бізнес серед конкурентів.

4) Виділятися серед конкурентів: веб-сайт з прокату автомобілів із привабливим дизайном, зручною навігацією та цінним контентом може виділити ваш бізнес з прокату автомобілів серед інших у цьому регіоні. Забезпечуючи безперебійний онлайн-досвід, ви можете привернути увагу потенційних клієнтів та продемонструвати унікальну цінність, яку пропонує ваш бізнес порівняно з іншими.

Вартість створення веб-сайту прокату автомобілів може змінюватися залежно від необхідних вам функцій та можливостей. Наприклад, створення веб-сайту за допомогою Wix починається від 14 доларів на місяць для базових планів. Однак, можливо, доведеться врахувати додаткові витрати на індивідуальний дизайн, розробку та інтеграцію сторонніх інструментів або плагінів. Використання веб-дизайнера також може значно збільшити вартість вашого веб-сайту.

Щоб захистити дані користувачів на веб-сайті прокату автомобілів, треба використовувати надійні заходи безпеки, такі як SSL-сертифікати, для шифрування даних між вашим сайтом та користувачами, захищаючи платіжні дані. Регулярно оновлювати програмне забезпечення та плагіни

вашого веб-сайту, щоб уникнути вразливостей. Використовувати безпечні платіжні шлюзи, що відповідають стандартам PCI DSS для транзакцій. Надати чітку політику конфіденційності веб-сайту, щоб інформувати клієнтів про використання та зберігання даних, гарантуючи їм відповідальне поводження з ними. Регулярні аудити безпеки та моніторинг також можуть виявити загрози та посилити безпеку вашого веб-сайту.

Забезпечувати безперебійну роботу веб-сайту з прокату автомобілів, своєчасно перевіряючи та виправляючи непрацюючі посилання. Відстежувати такі аспекти продуктивності, як швидкість завантаження та взаємодія з користувачем. Переглядайте веб-аналітику, щоб зрозуміти поведінку відвідувачів та знайти області для покращення. Оперативно реагуйте на відгуки чи скарги клієнтів, щоб підтримувати свою репутацію.

1.2 Аналіз відомих програм аналогів

Проведемо аналіз відомих аналогів веб-ресурсу компаній прокату автомобілів [6-19].

Для якісної реалізації веб-ресурсу служби оренди автомобілів доцільно здійснити аналіз популярних існуючих платформ, які вже функціонують на ринку та демонструють ефективність. Це дозволяє виявити найкращі практики, функціональні рішення, а також типові недоліки, яких слід уникати під час створення власного ресурсу. Більш детально розглянемо деякі із них для визначення основних показників юзабіліті, зрозумілості, структури, доступу до послуг.

1. Rentalcars. Один із найпопулярніших міжнародних сервісів з бронювання автомобілів [6].

Основні переваги платформи:

- інтуїтивно зрозумілий інтерфейс;

- гнучка система фільтрації за класом авто, типом коробки передач, ціною та іншими параметрами;
- підтримка багатьох мов та валют;
- інтеграція з системами онлайн-оплати;
- наявність мобільного додатку.

Недоліком можна вважати

- перевантаження додатковими пропозиціями, що ускладнює процес вибору для нових користувачів;
- не завжди прозора інформація щодо додаткових платежів.

2. Economy Bookings. Цей сервіс має широку географію надання послуг і дозволяє орендувати авто в понад 150 країнах [7].

Сильні сторони:

- проста процедура оформлення бронювання;
- можливість порівняння цін від різних прокатних компаній;
- система бонусів для постійних клієнтів;
- надання додаткових послуг (страхування, GPS, дитячі крісла тощо).

Недолік - користувачі часто відзначають певну непрозорість у формуванні кінцевої вартості оренди.

3. Uber Rent та Lyft Rentals. Це сервіси, що поєднують класичне таксі з можливістю орендувати транспортний засіб [8, 9].

Переваги:

- інтеграція з уже знайомими мобільними додатками;
- швидка процедура замовлення;
- прозоре ціноутворення;
- зручне керування орендою через додаток.

Недолік - обмеження у географічній доступності, функція доступна переважно у великих містах США та Європи.

4. Localrent. Цей сервіс орієнтований на ринки Східної Європи, Кавказу та Азії [10]. Його відмінність - співпраця з локальними прокатними фірмами.

Особливості сервісу:

- конкурентні ціни;
- гнучкі умови бронювання;
- прямий контакт з місцевими постачальниками послуг;
- відсутність необхідності кредитної картки у деяких регіонах.

Основною проблемою може стати неоднорідна якість обслуговування в залежності від конкретного партнера.

- часто обмежений автопарк.

5. NarsCars. (регіональний український сервіс) [11]. Сервіс здебільшого орієнтовано на туристів та локальних мешканців, що потребують авто на короткий термін. Сайт має базову функціональність: вибір авто, календар оренди, онлайн-заявка.

Переваги:

- зрозуміла та проста структура сайту;
- пряме спілкування з менеджером;
- швидке оформлення.

Недоліки:

- відсутність автоматизації;
- вузький вибір автомобілів;
- обмежена мобільна версія або її повна відсутність.

6. Avis. Однієї із найбільших компаній з оренди автомобілів [12]. Орієнтована як на корпоративних, так і на індивідуальних клієнтів.

Переваги:

- Проста процедура бронювання;
- Програма лояльності для постійних клієнтів;
- Детальні описи авто;

- Сервіси для бізнес-користувачів.

Недоліки: обмеженість локальних пропозицій у деяких регіонах; складна система цін у поєднанні зі страховими опціями.

6. Uklon AutoRent. Місцевий сервіс прокату, що адаптований до внутрішнього ринку України [13]. Пропонують як погодинну, так і подовову оренду транспортних засобів з можливістю самостійного вибору місця отримання та повернення авто.

Переваги:

- інтеграція з мобільними застосунками;
- гнучкі тарифи;
- прив'язка до геолокації;
- мінімальний набір документів для оформлення.

Недоліки: менше охоплення території, ніж у міжнародних сервісів.

У таблиці 1.1 надано порівняння сайтів міжнародних компаній оренди авто з точки зору зручності користувача (юзабіліті, зрозумілість, структура, доступ до послуг):

- Uber Rentals. URL: <http://uber.com/rentals> ;
- Lyft Rentals. URL: <http://lyft.com/rentals> ;
- Localrent. URL: <http://localrent.com> ;
- Avis International. URL: <http://avis.com> ;
- Uklon. URL: <http://uklon.com.ua> ;
- Getmancar. URL: <http://getmancar.com> ;
- NarsCars. URL: <http://narscars.com.ua> .

Аналіз даних (табл.1.1) дозволяє зробити загальні висновки:

- найзрозуміліше подання інформації - Localrent, Uber Rentals;
- найбільш користувацько-дружній сайт для України - NarsCars, Getmancar;
- найменш орієнтовані на оренду авто сайти - Uklon, Lyft (фокус — не класична оренда);

- проблеми з доступністю інформації - Avis (глибока навігація), Lyft (маркетинг переважає інформативність).

Таблиця 1.1 Порівняння сайтів міжнародних компаній оренди авто

Назва сервісу	Переваги	Недоліки
Uber Rentals	• мінімалістичний, зрозумілий інтерфейс; • добре адаптований для мобільних користувачів; • ясне позиціонування, оренда "погодинно"	• обмежене пояснення умов; • інформація доступна лише англійською; • лише для певних країн
Lyft Rentals	• інформативна презентація у форматі блогу; • чіткі приклади використання; • пояснено переваги над традиційною орендою	• сторінка більше маркетингова, ніж функціональна; • немає прямої інтеграції з бронюванням; • обмежено географічно
Localrent	• дуже проста навігація; • вказані ціни, страхування, фото авто; • є українська мова та багато інших; • підтримка локальних орендодавців	• інтерфейс трохи застарілий візуально; • мало інформації про умови оплати/заставу на першому етапі
Avis International	• професійний, корпоративний стиль; • чітко виділені акційні пропозиції; • міжнародне покриття	• досить складна структура, багато підсторінок; • обмежений вибір мов; • вимагає багато кліків для доступу до реальних цін
Uklon	• орієнтація на українського користувача; • сучасний адаптивний дизайн • добре інтегрований сервіс для замовлення авто	• основний -таксі, а не оренда авто; • оренда або довгострокова аренда не на першому плані
Getmancar	• добре описані умови каршерінгу; • інтерактивна карта авто; • мобільна інтеграція через додаток	• складні умови для нових користувачів; • не завжди ясно, що входить у вартість; • слабка навігація сторінкою
NarsCars	• повна інформація про авто, ціни, умови; • зручний фільтр авто по параметрам; • є калькулятор вартості	• візуально перевантажений інтерфейс; • деякі елементи складні для мобільного перегляду; • частково застарілий дизайн

Перелік веб-ресурсів популярних служб прокату автомобілів у м.Вінниці, які надають послуги оренди авто з різними умовами, типами транспорту та рівнями сервісу [11-19] наведено нижче.

1. NarsCars. URL: <https://narscars.com.ua/ua> .
2. Avis. URL: <https://www.avis.com/en/home> .
3. Uklon. URL: <https://uklon.com.ua/> .
4. Rental.ua. URL: <https://rental.ua/ua/vinnitsa> .
5. 7Cars. URL: <https://7cars.com.ua> .
6. BLS Car Rental. URL: <https://bls.ua/ua> .
7. Car-Rent.ua. URL: <https://car-rent.ua> .
8. AutoProkat.net.ua. URL: <https://autoprokat.net.ua> .
9. Payless Car Rental. URL: <https://payless.com.ua> .

Порівняння інформативності сайтів оренди авто у Вінниці надано у таблиці 1.2.

Висновки: найкраще подання інформації на сайтах Rental.ua, 7Cars, Avis Україна; потребують покращення сайти AutoProkat.net.ua, Car-Rent.ua.

Отже, аналіз програм-аналогів показує, що успішні веб-ресурси для прокату авто об'єднують простоту користування, надійність, гнучкість у виборі авто та прозорість умов. Проте навіть великі сервіси мають певні недоліки - від перевантаженості функціями до обмеженої географії.

Проведений аналіз демонструє, що ефективний веб-ресурс служби прокату автомобілів повинен поєднувати такі ключові елементи:

- інтуїтивно зрозумілий і адаптивний інтерфейс;
- широкий вибір авто з фільтрацією за параметрами;
- наявність особистого кабінету користувача;
- прозора система оплати;
- мобільна версія або застосунок;
- підтримка онлайн-бронювання в реальному часі.

Таблиця 1.2 – Порівняння інформативності сайтів автопрокату у Вінниці

Сайт	Переваги	Недоліки
NarsCars	- простий та зрозумілий інтерфейс; - чітке представлення цін та умов оренди	- обмежена кількість фотографій авто; - відсутність детальної інформації про додаткові послуги
Avis Україна	- висока якість подання інформації; - детальні описи авто та умов оренди	сайт орієнтований на міжнародного користувача, що може ускладнити навігацію для місцевих клієнтів
Uklon	- інтеграція з мобільними застосунками; - гнучкі тарифи; - прив'язка до геолокації; - мінімальний набір документів для оформлення	менше охоплення території, ніж у міжнародних сервісів
Rental.ua	- чіткий та інтуїтивно зрозумілий інтерфейс– детальні описи авто з фотографіями; - зручна система фільтрації та пошуку	- обмежена інформація про додаткові послуги; - відсутність відгуків клієнтів на сайті
7Cars	- сучасний дизайн сайту; - повна інформація про умови оренди; - наявність блогу з корисними порадами	- деякі розділи потребують оновлення; - відсутність онлайн-чату для швидкого зв'язку
BLS Car Rental	- професійний вигляд сайту; - детальні описи авто з технічними характеристиками	- складна навігація для нових користувачів; - відсутність інформації про наявність авто у вінниці
Car-Rent.ua	- зручна система пошуку та фільтрації; - чітке представлення цін	- застарілий дизайн сайту; - відсутність мобільної версії сайту
AutoProkat.net.ua	- мінімалістичний дизайн; - швидке завантаження сторінок	- дуже обмежена інформація про авто; - відсутність фотографій та детальних описів
Payless Car Rental	- простий та зрозумілий інтерфейс; - чітке представлення цін та умов оренди	- обмежена кількість фотографій авто; - відсутність детальної інформації про додаткові послуги

Частково усі перераховані вимоги можна розв'язати з використанням наявних на ринку конструкторів веб сайтів, але повне врахування цих особливостей забезпечить тільки індивідуальна розробка, яка дозволить створити конкурентоспроможний продукт, орієнтований на зручність користувача та ефективну взаємодію з клієнтом.

1.3 Постановка задачі

Завдання, які ставляться при розробці сайту, являють собою набір конкретних дій, реалізація яких призведе до досягнення кінцевої мети. Завдання плануються в залежності від багатьох факторів - бюджету проекту, його важливості для компанії, наявних ресурсів для його реалізації і підтримки, часових обмежень тощо. Правильна постановка завдань є дуже важливим етапом.

Перед розробниками сайту, можуть ставитися різні завдання. Ось, основні із них [20]:

- вбудовані додаткові функції, наприклад, інтеграція з обліковими програмами, окремі модулі crm (систем управління взаємовідносинами з клієнтами) та erp (управління підприємницькими процесами). наприклад, автоматичний розподіл замовлень на доставку між кур'єрами інтернет-магазину;
- розробка зручного адміністративного інтерфейсу для керування ресурсами сайту: система управління контентом, контроль прав доступу та адміністрування користувачів;
- надавання клієнтам доступу до цифрових сервісів, зокрема оформлення покупок у мережових магазинах, бронювання квитків, оплата через електронні платіжні платформи;
- впровадження інструментів маркетингу — системи зворотного зв'язку, онлайн-анкет, опитувань, тестувань тощо;

- інтеграція платформ для комунікації користувачів: коментарі, форуми, особисті блоги, дошки оголошень, функція приватного листування;
- розробка корисних і зручних онлайн-сервісів для основної аудиторії — зокрема інформаційних, пошукових, розважальних тощо.
- створення унікального дизайну, що гармонійно поєднується з корпоративною айдентикою компанії.

На початковому етапі розробки вебресурсу першочергово визначаються найбільш критично важливі завдання, реалізація яких має найвищий пріоритет у рамках проєкту.

Усі вимоги, що формулює замовник під час створення сайту, повинні узгоджуватись із загальною стратегією компанії в галузі маркетингу та реклами.

Під час формування технічного завдання вирішується питання побудови структури вебсайту.

Під структурою, зазвичай, мається на увазі організація розділів ресурсу, які відображаються у вигляді пунктів навігаційного меню. Також на цьому етапі відбувається визначення функціонального наповнення сайту та формулювання вимог до його візуального оформлення [21, 22].

Сучасний користувач цінує передусім змістовне наповнення. Якщо вебсайт надає корисну, актуальну та якісну інформацію, неідеальне оформлення або наявність реклами не стають критичними факторами. Саме тому ресурси, які забезпечують релевантний контент навіть за умов простого дизайну, часто мають стабільну та високу відвідуваність.

Варто зазначити, що користувачі не читають вебсторінки послідовно — вони сканують їх, шукаючи візуальні маркери: заголовки, списки, ключові фрази, виділені шрифтом.

Також користувачі схильні до нетерпіння: вони очікують швидкої реакції ресурсу на свої запити. Якщо вебсайт не відповідає очікуванням цільової аудиторії, компанія ризикує втратити потенційний прибуток. Чим складніше користувачеві орієнтуватися на сайті, тим вища ймовірність, що він залишить його.

Користувачі зазвичай не прагнуть до пошуку оптимальних шляхів, їх поведінка в інтернеті часто є спонтанною. Вони не переглядають сторінки лінійно, а хаотично переміщують погляд, зупиняючись на елементах, які здаються релевантними. Зазвичай достатньо одного візуального сигналу, що може містити необхідну інформацію, аби користувач негайно клікнув на нього [21].

Ще один важливий аспект — потреба в контролі.

Люди не терплять зовнішнього втручання в роботу їх браузера: зміну розмірів вікна, неочікувану появу спливаючих вікон, ігнорування блокувальників тощо.

Перший візит на сайт не повинен викликати у користувача питання на кшталт: «Де я опинився?» або «Як знайти потрібну інформацію?».

Навігація повинна бути інтуїтивно зрозумілою, а доступ до потрібних розділів — максимально швидким і зручним. Якісний вебдизайн — це насамперед логічно організований, зручний та функціональний сайт.

1.4 Висновок до розділу 1

Створення власного WEB-ресурсу служби прокату автомобілів є достатньо актуальною задачею, розв'язання якої надає ряд переваг, а саме: наявність тільки необхідних функцій; індивідуальний дизайн; контроль трафіку; заощадження коштів. У якості завдання обираємо розробку WEB-ресурсу служби прокату автомобілів.

2 РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ WEB-РЕСУРСУ СЛУЖБИ ПРОКАТУ АВТОМОБІЛІВ

2.1 Розробка структури WEB-ресурсу

Розробка навігаційної структури одна з найважливіших складових, що впливають на створення сайтів.

У цьому блоці міститься все, що стосується вмісту сайту, а також інформаційної стратегії створюваного проекту. Це важливо, щоб користувачі могли швидко і без особливих труднощів знайти потрібну інформацію. Для цього продумується можливий шлях користувачів по сторінках сайту та розробляється навігація, яка повинна бути інтуїтивно зрозуміла кожному відвідувачу [20].

На основі структури буде побудована робота із створення інтернет - проекту. Схема буде доповнена текстовою інформацією, візуальними образами, програмними модулями і багатьо чим іншим, з чого складатиметься сайт. Професійно розроблена структура дає можливість значно скоротити витрати на збір інформативної складової сайту та уникнути великої кількості помилок на етапі виробництва. Такий сайт працює значно ефективніше і не потребує постійних виправлень в процесі його роботи.

В інтернеті існує безліч сайтів з корисною інформацією, але проблема полягає в тому, що цю інформацію складно знайти, через те, що сайт має незрозумілу, або занадто заплутану структуру. Найбільш часто помилками навігаційної структури, що зустрічаються, є: надто заплутане меню, неправильне розміщення важливих сторінок, великі рівні вкладеності сторінок, незрозуміле та хаотичне розташування інформації.

Створення оптимальної структури - дуже важлива складова створення сайту. Якщо користувач зайдет на перевантажену інформацією сторінку, то, швидше за все, він просто її закриє. Сторінку необхідно створити такою, що

б моментально можливо було визначити її вміст і знайти необхідну інформацію або одне з посилань [20].

Як показує досвід, у замовника є зібрана інформація, яку він би хотів розмістити на своєму сайті, але, в більшості випадків, ця інформація підлягає структуризації і вимагає розбиття на частини та розділи. У свою чергу всі статті та розділи вимагають створення оптимальної системи посилань, для спрощення користування сайтом і скорочення часу на пошук необхідної інформації.

Отримавши всю необхідну інформацію для майбутнього сайту (тематика й завдання сайту, обсяг інформації і співвідношення текстових і графічних компонентів), наші фахівці в області розробки сайтів проводять аналіз та пропонують оптимальну навігаційну структуру для майбутнього сайту. Завдяки цьому, користувачі вашого сайту зможуть швидко знайти цікаву для них інформацію, користуючись інтуїтивно зрозумілою та зручною структурою сайту. Тому, перед створенням інтернет-сайту, необхідно проводити комплексний аналіз з дослідження інтернету на певну тематику, і виявити переваги та недоліки аналогічних ресурсів.

Та легкість, з якою сучасні HTML-редактори дозволяють створювати гіперпосилання на основі довільного фрагмента тексту документа або графічного зображення, викликають спокусу розставити їх скрізь, де лише можна. Проте практика показує, що надмірне захоплення створення заслань може вийти боком як творцеві сайту, так і відвідувачам.

Загальні правила проектування навігаційної схеми сайту

Для виконання приведеної вище вимоги досить дотримувати нескільки не дуже складних правил:

на головній сторінці сайту обов'язково має бути представлена карта сайту, причому не лише в графічній, але і в текстовій формі (детальніше про структуру головної сторінки ми поговоримо трохи пізніше);

на всіх подальших сторінках мають бути присутніми заслання на головну сторінку і на першу сторінку даного розділу (на якій, у свою чергу, має бути представлений вміст розділу);

при використанні малюнків як заслання вони мають бути однотипними на всіх сторінках і розташовуватися в одній і тій же позиції;

малюнки, використовувані як заслання (у тому числі «кнопки»), мають бути інтуїтивно помітні для читачів відносно інших елементів сторінки.

На досить крупних сайтах додатково доцільно передбачити можливість оперативного зв'язку з автором (Web-мастером), до якого можна було б звернутися в скрутних ситуаціях [21].

Для реалізації навігаційної схеми зазвичай використовуються наступні типи заслань (текстових або графічних):

внутрішні заслання, що зв'язують сторінки в межах одного сайту;

зовнішні заслання, що забезпечують зв'язок з документами, розташованими поза даним сайтом;

заслання на адреси електронної пошти, які відкривають форму для відправки електронної пошти;

локальні заслання (інколи іменовані якорями), які дозволяють відвідувачеві переходити на конкретний фрагмент тієї ж сторінки або одній з інших сторінок вузла.

Продумана навігаційна схема сайту повинна надавати відвідувачам важливу можливість - оцінити, яка інформація знаходиться на вузлі, наскільки швидко можна до неї добратися. Знаходячись на будь-якій сторінці вузла, відвідувачі повинні представляти своє місце розташування, знати, що вони можуть зробити на цій сторінці, і куди можуть піти далі.

Крім того, якісно продумана і організована структура позитивно впливає на оптимізацію сайту в пошукових системах. Існує каскадна і деревоподібна структури сайтів. В каскадному випадку посилання в документах задані таким чином, що існує тільки один шлях обходу сторінок

веб-сайту. За каскадного способу організації сторінок відвідувачі сайту можуть переміщуватися тільки в одному з напрямків — вперед або назад

Масштабованість - це важлива властивість будь-якого сайту: яким би високим не був первинний рейтинг, вузол обов'язково повинен розвиватися. Застиглі форми чужі самій природі Інтернету. Необхідність розвитку вузла може бути обумовлена і «внутрішніми» причинами: зростанням професійної майстерності його творця, збільшенням об'єму послуг, що надаються, організацією - власником вузла, прагненням до підвищення конкурентоспроможності і так далі [22].

Крім того, масштабованість вузла є одним з умов узгодженості його інтерфейсу: після того, як відвідувачі вже вивчили структуру і схему навігації вузла, вони спиратимуться на це знання під час подальших відвідин. Якщо в своїх пошуках інформації вони стануть втрачати орієнтацію і потрапляти зовсім не туди, куди звиклися, наступних відвідин може і не бути.

На рисунку 2.1 надано каскадну структуру сайту.

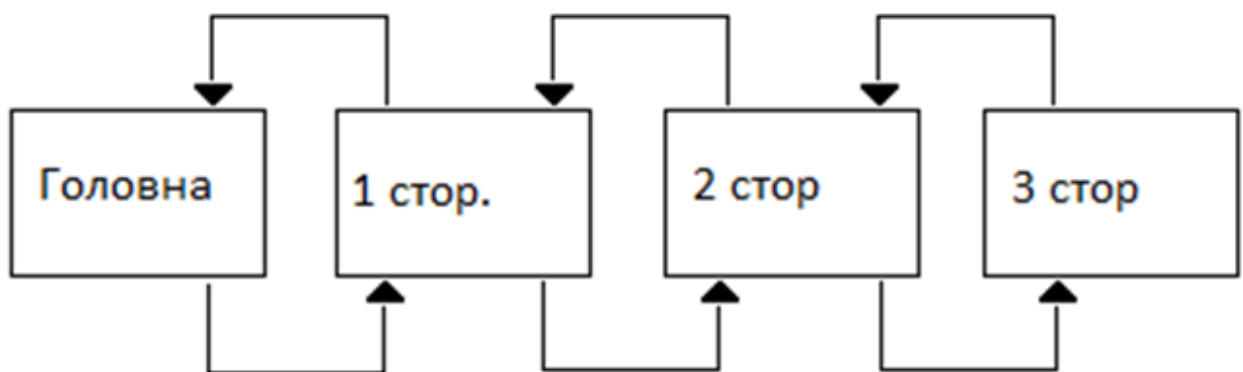


Рисунок 2.1 – Каскадна структура сайту

На сайті з деревоподібною структурою з головної сторінки можна потрапити на одну зі сторінок другого рівня, звідти — на одну зі сторінок третього рівня тощо (рис.2.2).

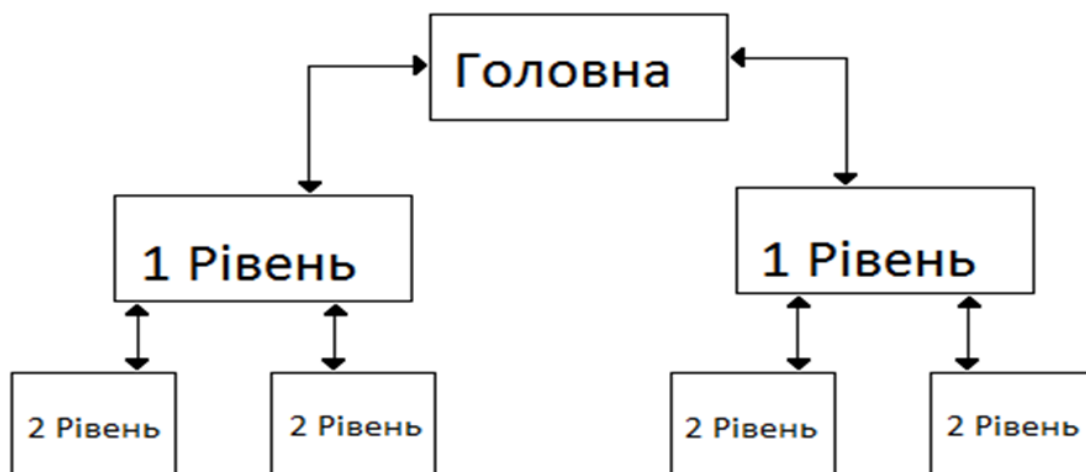


Рисунок 2.2 – Деревоподібна структура сайту

Для реалізації сайту краще обрати вільну структуру тому що вона є найбільш розповсюджена і найбільш зручна для користувача (рис.2.3).

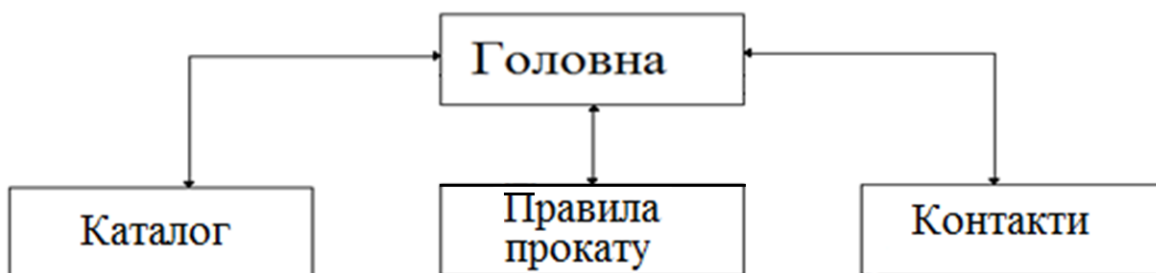


Рисунок 2.3 – Вільна структура сайту

На рисунку 2.4 надано загальний вигляд сторінки сайту.

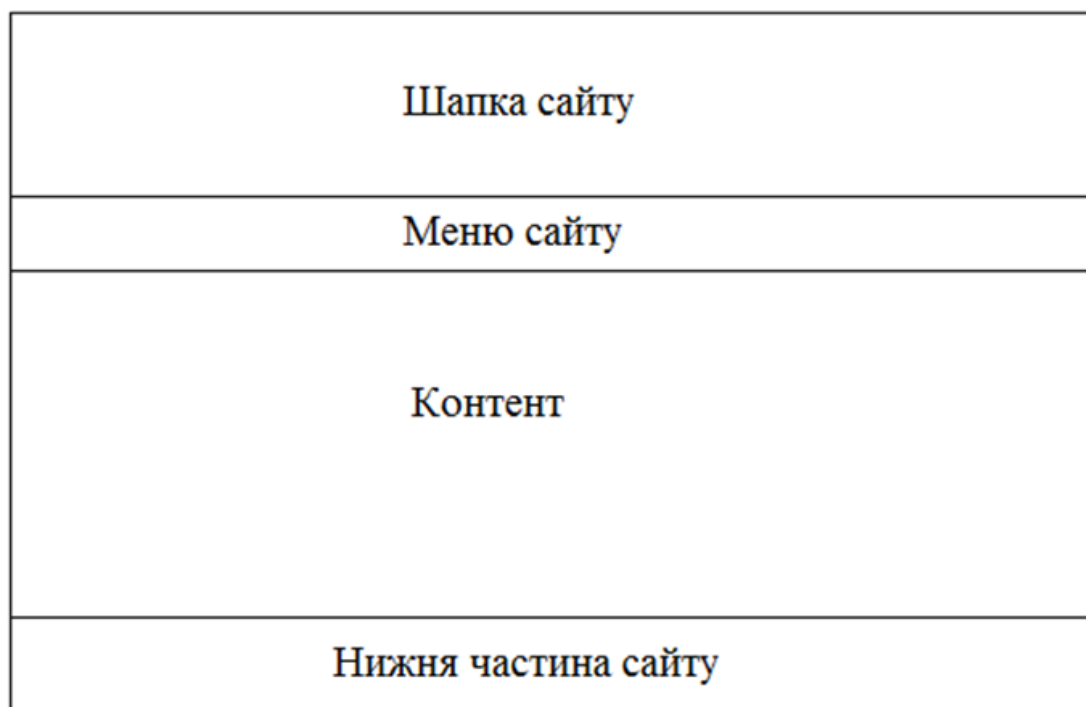


Рисунок 2.4 – Вигляд сторінки сайту

2.2 Розробка діаграми класів WEB-ресурсу служби прокату автомобілів.

Опис акторів-користувачів: «Користувач» - користувач інформаційної системи для оренди авто; «Адміністратор» - користувач системи у ролі «адміністратора».

UML-діаграма варіантів використання наведена рисунку 2.5.

Зміст варіантів використання надано у таблиці 2.1.

На рисунку 2.6 надано UML-діаграму прецедентів “Бронювання автомобіля”.

На рисунку 2.7 надано UML-діаграму прецедентів “Авторизація адміністратора”.

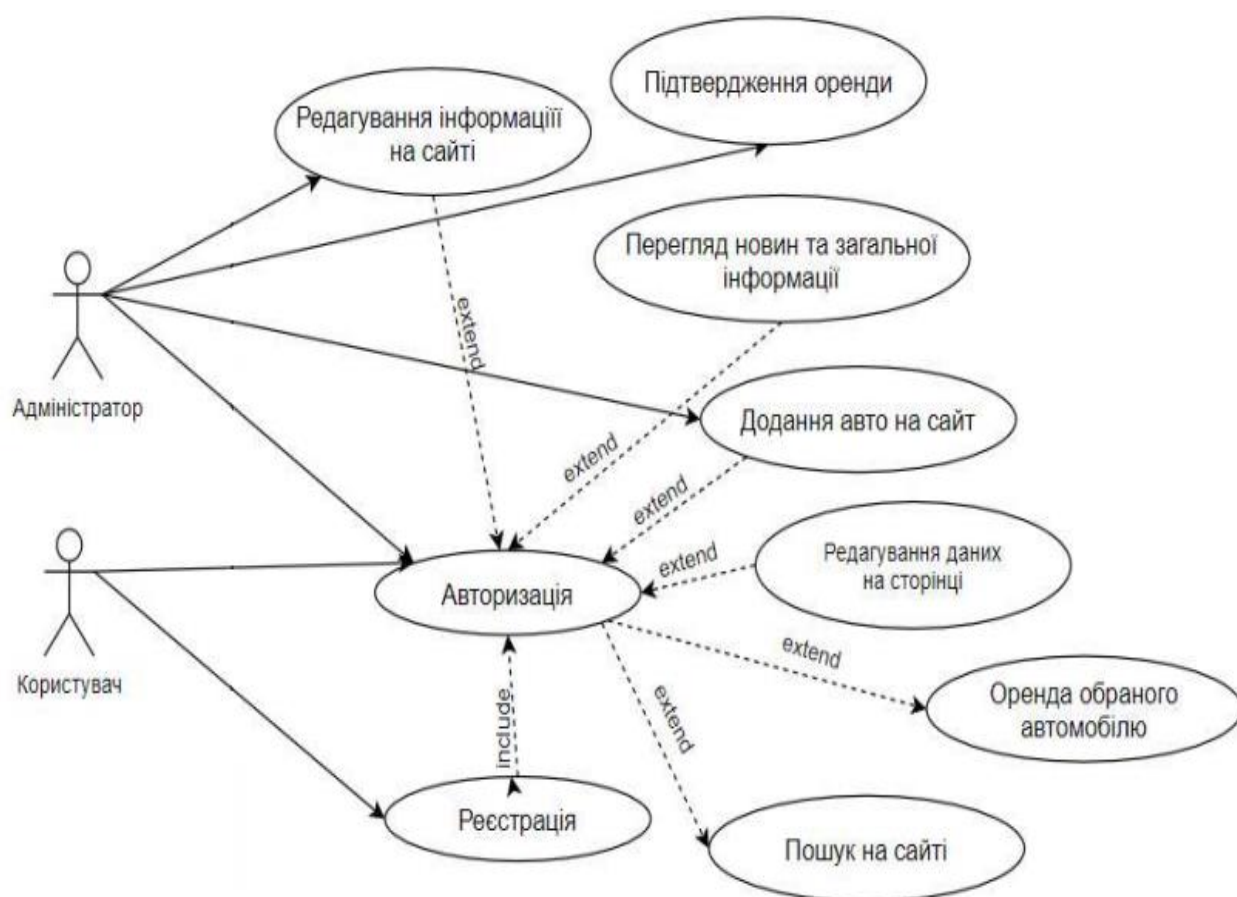


Рисунок 2.5 – UML-діаграма варіантів використання

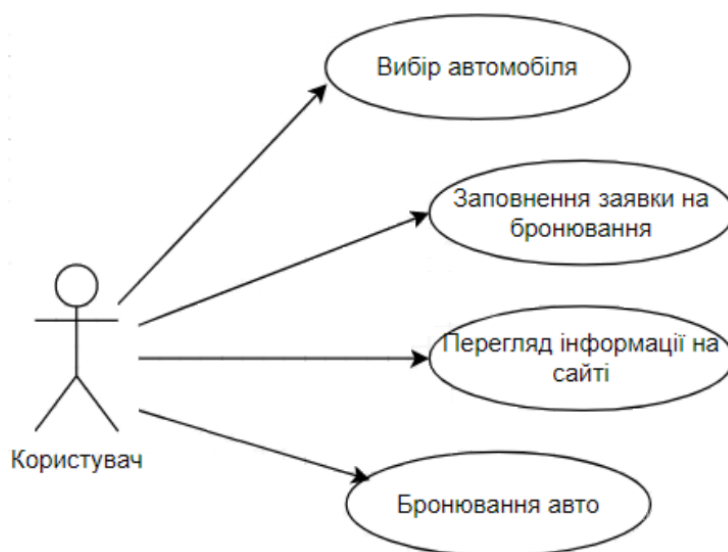


Рисунок 2.6 - UML-діаграма прецедентів “Бронювання автомобіля”

Таблиця 2.1 - Зміст варіантів використання

Назва	Опис
Авторизація	Функція входу в систему.
Реєстрація	Користувач має можливість реєстрації у системі.
Редагування інформації на сайті	Адміністратор має можливість редагувати дані на сайті.
Додання авто на сайт	Адміністратор може додати інформацію про нові автомобілі на сайт.
Підтвердження оренди	Адміністратор може підтверджувати запит на оренду авто.
Перегляд новин та загальної інформації	Перегляд користувачами інформації на сайті.
Редагування даних на сторінці	Можливість редагування користувачем власних даних.
Пошук на сайті	Користувач має можливість пошуку авто з використанням фільтру
Оренда обраного автомобілю	Можливість оформлення запиту на оренду обраного автомобілю.



Рисунок 2.7 - UML-діаграма прецедентів “Авторизація адміністратора”

UML-діаграма класів описує логічну структуру для WEB-ресурсу служби прокату автомобілів (рис.2.8).

Основні об'єкти, які можна виокремити:

- Car — автомобіль, головний клас, містить усі технічні характеристики автомобіля;
- RentalPrice — ціна оренди за певну кількість днів, описує ціни оренди залежно від кількості днів (1-4, 5-14 тощо);
- Color — колір, виділений окремо для збереження кольору у вигляді hex-коду (#ffffff тощо) ;
- Image — зображення;
- Order — можлива майбутня сутність (натиснення кнопки "замовити");
- getRentalPrice(days: int) — метод, який би дозволяв у програмі визначати, скільки коштує оренда за певну кількість днів.

Customer (Клієнт) має такі атрибути та методи.

Атрибути:

- customerId: int;
- name: String;
- phone: String;
- email: String;
- driverLicenseNumber: String.

Методи:

- register();
- updateProfile();
- viewOrders().

Order (Оренда) має такі атрибути та методи.

Атрибути:

- orderId: int;
- customer: Customer;

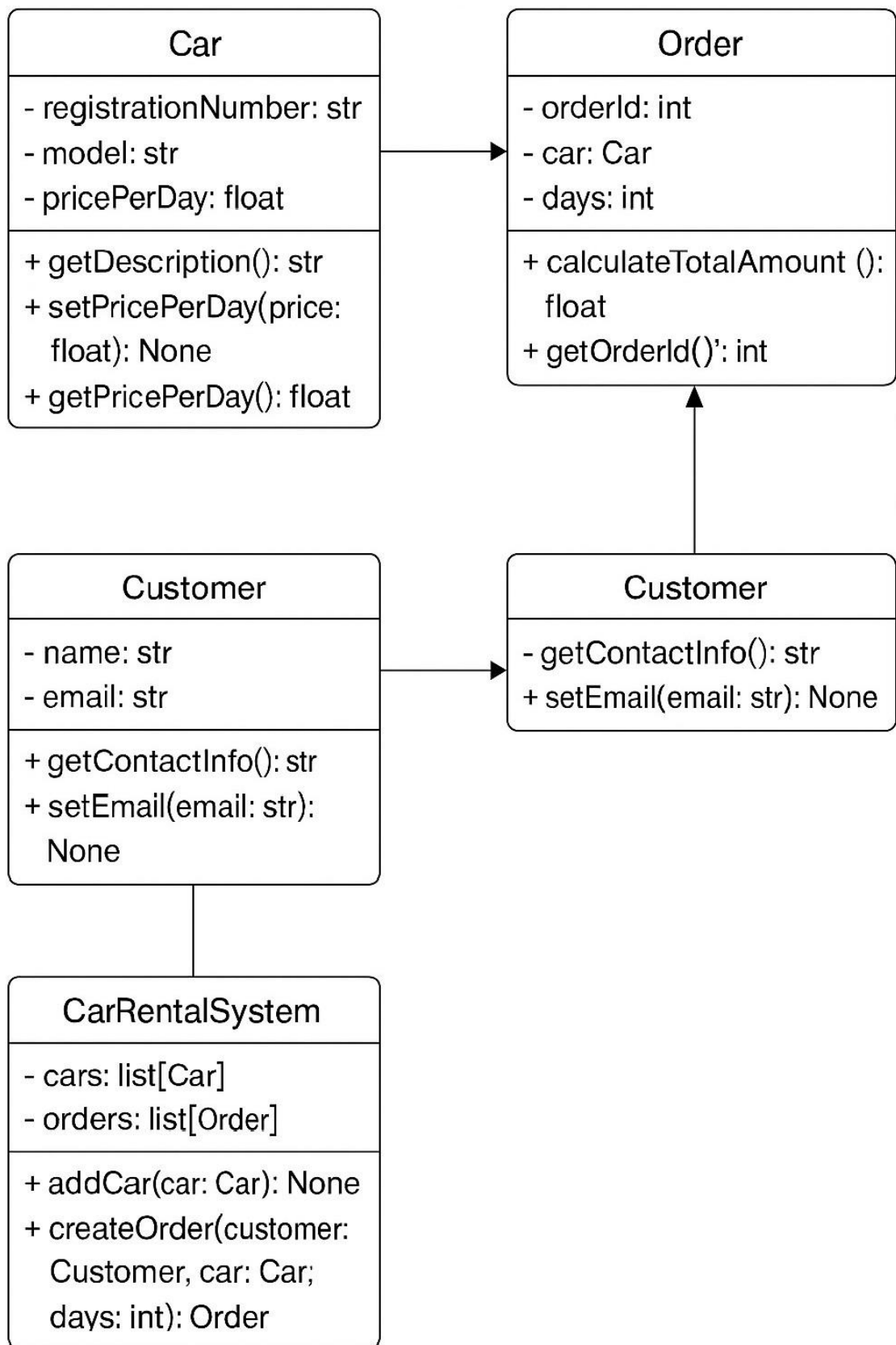


Рисунок 2.8 - UML-діаграма класів

- car: Car;
- startDate: Date;
- endDate: Date;
- totalPrice: double;
- status: String.

Методи:

- calculatePrice();
- confirmOrder();
- cancelOrder().

Catalog (Список авто) має такі атрибут та методи.

Атрибут:

- cars: List.

Методи:

- searchByModel(model: String);
- filterByAvailability();
- sortByPrice().

Основні зв'язки між класами:

- Customer — Order (один клієнт може мати багато оренд);
- Order — Car (кожна оренда стосується одного авто);
- Catalog — Car (каталог містить багато авто).

2.3 Послідовність дій WEB-ресурсу служби прокату автомобілів

Для побудувати UML діаграму послідовності дій WEB-ресурсу служби прокату автомобілів, слід розглянути взаємодію між користувачем та елементами сторінки, особливо під час перегляду інформації про авто та натискання кнопки “замовити” (рис.2.9).

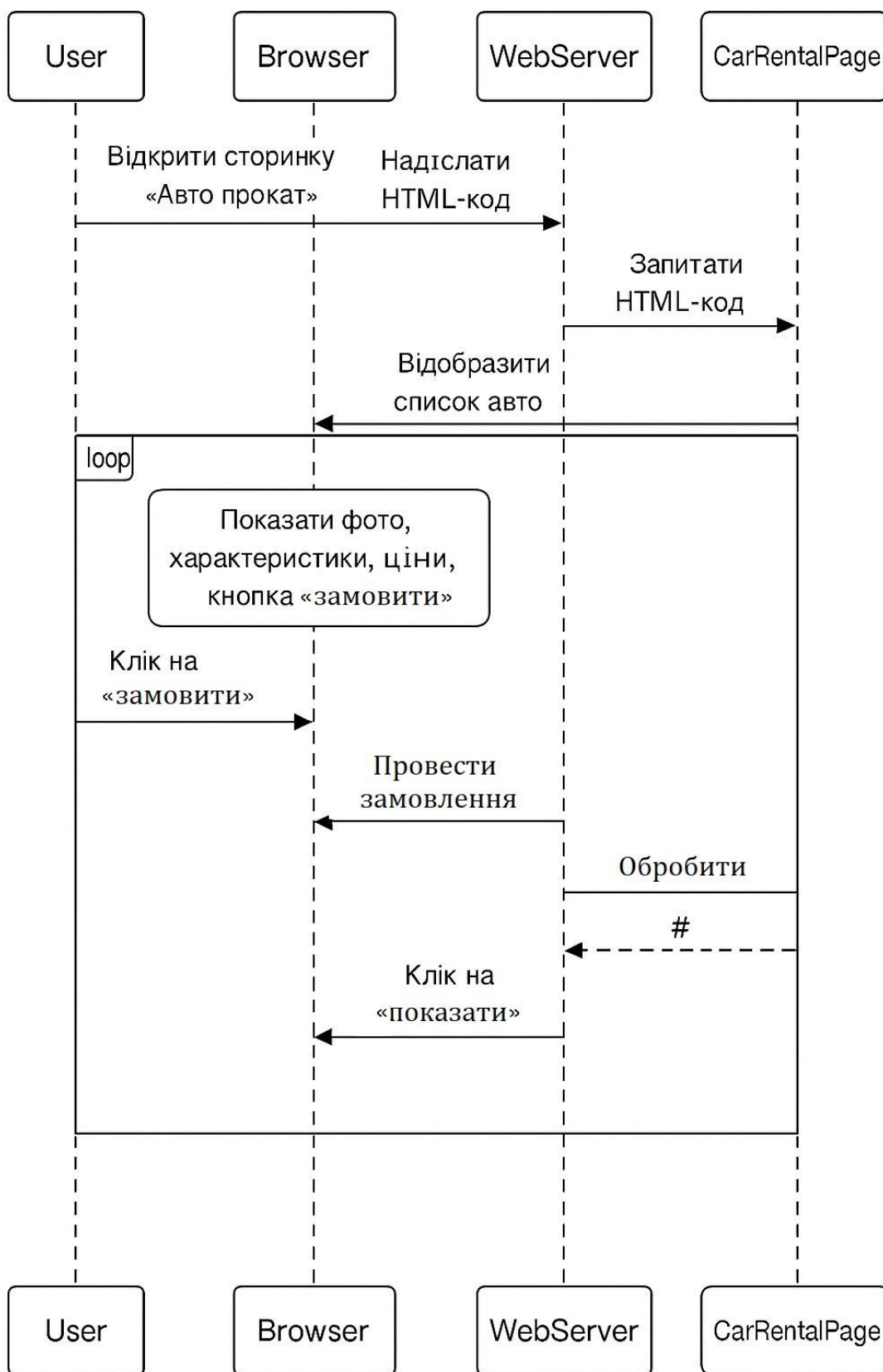


Рисунок 2.9 - UML діаграму послідовності дій WEB-ресурсу
служби прокату автомобілів

Основні дії зводяться до виведення інформації про авто та статичного кліку по посиланню.

Основні об'єкти для діаграми такі.

1. Користувач (User) – відвідує сайт.
2. Веб-браузер (Browser) – завантажує HTML/CSS.
3. HTML-сторінка (CarRentalPage) – відображає контент (машини, характеристики, ціни).
4. Сервер (WebServer) – надає HTML-сторінку та ресурси (зображення, CSS, JS).
5. Сценарій “Замовлення” – посилання на іншу сторінку/обробник.

Діаграма відображає процес завантаження сторінки, виведення списку авто та пасивний клік по посиланню "замовити", яке веде до логіки обробки замовлення.

2.4 Висновок до розділу 2

У результаті досліджень розроблена структура WEB-ресурсу служби прокату автомобілів, визначено основні класи для програмування ресурсу, обрано структуру головної сторінки сайту.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ WEB-РЕСУРСУ СЛУЖБИ ПРОКАТУ АВТОМОБІЛІВ

3.1 Обґрунтування вибору мови програмування

Веб-розробка перетворилася з суто технічної галузі на динамічну та швидкозростаючу. Ця еволюція підкреслює важливість розуміння мов програмування, що використовуються у веб-розробці, таких як JavaScript та Python.

Ці мови служать мостом між кодом, зрозумілим для людини, та машинно-виконуваними інструкціями. Однак, під час розробки будь-якого програмного застосунку необхідно ретельно вибрати найкращу мову для веб-розробки, яка відповідає потребам проекту [23].

Мови програмування перетворили веб-розробку з технічного завдання розмітки та кодування на творче заняття. Вони дозволяють розробникам втілювати в життя свої ідеї, пов'язані з програмним забезпеченням.

Це перекладачі, які перетворюють інструкції розробників у форму, зрозумілу для комп'ютерів. Ці мови програмування мають організовані правила та формати, яких слід дотримуватися для правильної роботи коду. Хоча деякі новачки можуть плутати мови програмування з мовами кодування, це різні поняття.

Він охоплює формальні системи для комп'ютерного спілкування, що дозволяє розробникам писати код, який можуть зрозуміти комп'ютери. Мови програмування є більш неформальними та стосуються конкретних мов, що використовуються для написання коду. Він дає комп'ютерам інструкції виконувати певні завдання, пишучи інструкції мовами програмування.

Мови програмування мають різні важливі функції, до яких належать наступні:

- синтаксис - правила та структура, що використовуються для

композиції коду мовою програмування;

- типи даних - категорії значень, які підтримує програма, такі як числа, рядки та логічні значення;
- змінні - призначені місця пам'яті, здатні зберігати значення;
- оператори - символи виконують операції зі значеннями, такі як додавання, віднімання та порівняння;
- структури керування - оператори, що знаходять потік програми, включаючи умови if-else, цикли та виклики функцій;
- бібліотеки та фреймворки - збірки попередньо написаного коду, що спрощують повсякденні завдання та пришвидшують розробку;
- парадигми - стиль програмування або філософія, що застосовується в мові, наприклад, процедурна, об'єктно-орієнтована або функціональна.

Вибір правильної мови програмування для веб-розробки може суттєво вплинути на швидкість розробки та інтеграцію важливих функцій. Вибираючи мову програмування, важливо враховувати інші аспекти, такі як розмір спільноти розробників, наявність репозиторіїв для отримання інформації та простота початку роботи.

Крім того, переконайтеся, що мова програмування відкрита для інтеграції з іншими платформами для тестування. Ці критерії є вирішальними в тестуванні програмного забезпечення, забезпечуючи функціональність та сумісність додатків на різних пристроях, браузерах та операційних системах.

Під час проведення автоматизованого тестування доцільно використовувати відповідні фреймворки для автоматизованого тестування, які забезпечують високий рівень контролю коду та точну документацію та представлення результатів тестування.

Вибір системи автоматизованого тестування, яка дозволяє виконувати тестові випадки, як у веб-додатках, так і в мобільних додатках, є критично

важливим. Для досягнення цієї мети розгляньте можливість використання хмарних платформ тестування, які підтримують широкий спектр браузерів, пристроїв та платформ. Однією з таких платформ є LambdaTest.

LambdaTest — це платформа для виконання тестів на основі штучного інтелекту, яка спрощує тестування у великих фермах з понад 5000 комбінацій браузерів та ОС. Її хмарна інфраструктура на реальних пристроях дозволяє виконувати як ручні, так і автоматизовані тести, забезпечуючи швидше розгортання та масштабування за допомогою тестування мобільних додатків та кросбраузерного тестування.

Веб-розробка охоплює низку дисциплін, пов'язаних з проектуванням та управлінням веб-сайтами. Вона передбачає використання інструментів та фреймворків для створення веб-сторінок, які є адаптивними, інтерактивними та візуально привабливими. Веб-розробка поділяється на дві основні категорії: фронтенд-розробка, яка зосереджена на користувацькому інтерфейсі, та бекенд-розробка, яка обробляє серверні процеси.

Список найкращих мов програмування для веб-розробки у 2025 році включає JavaScript, Python, HTML, CSS, PHP, C#, Swift, Kotlin, Go, TypeScript, SQL, Rust, Scala, Perl, R, Dart та MATLAB, серед інших [23]. Ці мови відповідають вимогам сучасної веб-розробки, дозволяють створювати масштабовані, ефективні та динамічні веб-додатки.

Розглянемо коротку характеристику напопулярніших із них [24].

HTML. Мова розмітки, яка визначає структуру веб-сторінок, включаючи текст, зображення та посилання.

CSS. Мова таблиць стилів, що використовується для керування макетом, зовнішнім виглядом та дизайном веб-контенту, такого як кольори та шрифти.

JavaScript. Мова програмування, що використовується для додавання інтерактивності веб-сайтам, що дозволяє створювати динамічний контент, такий як форми, анімацію та оновлення в режимі реального часу.

Java. універсальна мова програмування, що використовується для створення великомасштабних веб-додатків та систем корпоративного рівня.

Python. Мова високого рівня, що використовується для веб-розробки, аналізу даних, автоматизації та машинного навчання.

PHP. Серверна мова сценаріїв, що використовується для створення динамічного контенту на веб-сторінках та взаємодії з базами даних.

C#. Мова програмування, призначена для створення веб-застосунків та сервісів, зокрема, з використанням .NET framework.

Ruby. Мова програмування, що використовується для розробки вебзастосунків за допомогою фреймворку Ruby on Rails.

Swift. Сучасна мова програмування, розроблена Apple, яка в основному використовується для створення додатків для iOS та macOS.

Kotlin. Мова, що покращує розробку для Android, також застосовується для створення веб-застосунків на стороні сервера.

Perl. Мова сценаріїв, яка традиційно використовується для веб-розробки, обробки тексту та завдань системного адміністрування.

.NET. програмний фреймворк, який підтримує створення веб-застосунків, API та сервісів, зазвичай з використанням таких мов, як C#.

GoLang. Мова програмування, створена для розробки швидких, ефективних та масштабованих бекенд-систем і веб-сервісів

Деякі з найпопулярніших мов веб-розробки - це HTML, CSS, JavaScript, Java, Python, PHP, Ruby, Swift та Perl, а також інші популярні мови веб-розробки, такі як .NET, C#, Kotlin, GoLang тощо.

Вибір найкращої мови веб-розробки у 2025 році залежить від конкретних вимог проекту, існуючого технологічного стеку, досвіду команди розробників та потреб масштабованості/продуктивності. Кожна мова має свої сильні сторони та варіанти використання.

Широко використовувана комбінація фронтенду включає HTML, CSS та JavaScript, причому JavaScript є дуже затребуваною та універсальною

мовою. Володіння JavaScript дозволяє працювати як з фронтенд-, так і з бекенд-технологіями [24].

Проаналізуємо основні можливості HTML, CSS та JavaScript [23-32].

HTML.

HTML слугує основою веб-розробки, забезпечуючи стандартизований підхід до створення веб-сторінок [25-27]. Тому оволодіння HTML є важливим для кожного веб-розробника. Він визначає структуру веб-сторінки та диктує її візуальне представлення.

Особливості HTML.

1. Семантичні елементи: HTML надає семантичні елементи, такі як `<header>`, `<footer>`, `<nav>` та `<article>`, щоб надати сенсу контенту, що покращує доступність та пошукову оптимізацію (SEO).
2. Гіперпосилання: Дозволяє створювати гіперпосилання (елементи `<a>`) для зв'язку різних веб-сторінок або ресурсів разом.
3. Функції доступності: Надає такі функції, як атрибути `alt` для зображень (`alt="description"`), щоб зробити веб-контент доступним для користувачів з інвалідністю.
4. Сумісність між браузерами: HTML підтримується всіма основними веббраузерами, що забезпечує однакову функціональність та відображення на різних платформах.
5. Адаптивний дизайн: підтримує принципи адаптивного дизайну за допомогою таких елементів, як теги `<meta>` та медіа-запити, що дозволяє веб-сторінкам адаптуватися до різних розмірів екранів та пристроїв.

Переваги HTML.

1. Простий та легкий у вивченні: HTML має зрозумілий синтаксис, що полегшує вивчення та розуміння початківцями.
2. Незалежність від платформи: веб-сторінки, написані мовою HTML, доступні з будь-якого пристрою з веб-браузером, незалежно від

операційної системи.

3. Широко підтримується: HTML підтримується всіма основними веббраузерами та є фундаментальною технологією Всесвітньої павутини.
4. Інтеграція з іншими технологіями: HTML можна поєднувати з CSS (каскадними таблицями стилів) для стилізації та JavaScript для інтерактивності, що забезпечує насичений веб-досвід.
5. SEO-дружність: Правильне використання семантичних елементів HTML може покращити рейтинг у пошукових системах, надаючи пошуковим системам більше контексту щодо вмісту веб-сторінки.

Обмеження HTML.

1. Статична природа: HTML в основному використовується для структурування контенту та не забезпечує динамічної поведінки, як мови програмування.
2. Обмежене стилізування: Хоча HTML може визначати структуру та вміст веб-сторінок, для візуального дизайну потрібне додаткове стилізування за допомогою CSS.
3. Проблеми доступності: Забезпечення повної доступності може вимагати додаткових зусиль, окрім базової HTML-розмітки, таких як надання відповідного тексту alt для зображень та реалізація належної семантичної структури.
4. Проблеми сумісності браузерів: Різні веббраузери можуть інтерпретувати HTML-код по-різному, що призводить до невідповідностей у відображенні та функціональності.

Випадки використання HTML.

1. Розробка веб-сайтів: HTML є основою розробки веб-сайтів, яка використовується для створення структури та вмісту веб-сторінок.
2. Шаблони електронних листів: HTML зазвичай використовується для створення шаблонів електронних листів для маркетингових кампаній

електронною поштою та розсилок.

3. Розробка веб-застосунків: HTML використовується разом з CSS та JavaScript для створення інтерактивних веб-застосунків та інтерфейсів користувача.
4. Платформи для блогів: HTML використовується на платформах для блогів для створення та форматування дописів і статей у блогах.
5. Онлайн-документація: HTML часто використовується для створення онлайн-документації та файлів довідки завдяки своїй простоті та зручності використання.

JavaScript.

JavaScript - це широко використовувана високорівнева інтерпретована мова програмування, яка є ключовою для створення та керування динамічним контентом на веб-сайтах [28-30]. Вона є фундаментальною технологією Всесвітньої павутини, що працює разом з HTML та CSS.

Особливості JavaScript.

1. Клієнтські сценарії: JavaScript в основному використовується для створення інтерактивного та динамічного контенту у веббраузерах.
2. Мова високого рівня: це мова програмування високого рівня, що робить код легким для написання та розуміння.
3. Динамічне типізування: Змінні не потребують явного оголошення типу даних, що дозволяє гнучке програмування.
4. ООП на основі прототипів: JavaScript використовує модель на основі прототипів для об'єктно-орієнтованого програмування.
5. Програмування на основі подій: підтримує програмування на основі подій, що дозволяє визначати обробники подій для дій користувача, таких як кліки та введення з клавіатури.

Переваги JavaScript.

1. Інтерактивність на стороні клієнта: покращує взаємодію з користувачем завдяки інтерактивним функціям, таким як перевірка

форм, анімація та динамічне оновлення вмісту без перезавантаження сторінки.

2. Універсальність: Використовується для веб-розробки, мобільних додатків (React Native), розробки на стороні сервера (Node.js) та розробки ігор (Phaser).
3. Багата екосистема: Велика екосистема бібліотек (jQuery, React, Vue.js) та фреймворків (Angular, Express.js), що спрощують розробку.
4. Підтримка спільноти: Велика та активна спільнота, яка робить внесок у проекти з відкритим кодом та надає значну підтримку й ресурси.
5. Швидкий цикл розробки: Динамічний характер та гнучкий синтаксис дозволяють швидке прототипування та ітерації, пришвидшуючи процес розробки.

Обмеження JavaScript.

1. Проблеми сумісності браузерів: Різні браузери можуть інтерпретувати JavaScript по-різному, що призводить до проблем сумісності та невідповідностей.
2. Вразливості безпеки: JavaScript може бути вразливим до проблем безпеки, таких як міжсайтовий скриптинг (XSS), якщо його належним чином не очистити та не перевірити.
3. Однопотокowe виконання: JavaScript виконується в одному потоці, що може спричинити вузькі місця в продуктивності в застосунках, що потребують інтенсивних обчислень або операцій вводу-виводу.
4. Пекло зворотних викликів: Асинхронне програмування зі зворотними викликами може призвести до складних, важкокерованих структур коду.
5. Проблеми налагодження: Налагодження JavaScript може бути складним через його динамічну природу та нечіткий типізацію, що ускладнює відстеження та вирішення помилок.

Випадки використання JavaScript.

1. Веб-розробка та дизайн: JavaScript є важливим для створення інтерактивних інтерфейсів користувача, перевірки форм на стороні клієнта та асинхронного зв'язку із сервером (AJAX).
2. Розробка мобільних додатків: такі фреймворки, як React Native, дозволяють розробникам створювати кросплатформні мобільні додатки за допомогою JavaScript та React.
3. Реалізація серверної логіки: середовища виконання JavaScript, такі як Node.js, дозволяють розробникам писати серверні додатки, API та мікросервіси.
4. Створення ігор: Такі фреймворки, як Phaser та Three.js, використовуються для розробки браузерних ігор та інтерактивної 3D-графіки.
5. Розробка настільних додатків: JavaScript у поєднанні з такими фреймворками, як Electron, може створювати кросплатформні настільні додатки за допомогою веб-технологій.

Фреймворки для веб-розробки на основі JavaScript включають React, Angular, Vuejs, Express.js, Next.js, Nuxt.js, Svelte, Meteor, Gatsby, Backbone.js тощо.

CSS.

CSS розшифровується як “Каскадні Таблиці Стилів” (Cascading Style Sheets) [27, 31, 32]. Це мова таблиць стилів, яка використовується для опису представлення документа, написаного на HTML або XML. CSS контролює, як елементи повинні відображатися на екрані, на папері, в мовленні або на інших носіях. Це дозволяє веб-розробникам створювати візуально привабливі веб-сторінки з різними макетами, кольорами та шрифтами.

Особливості CSS.

1. Візуальний стиль: За допомогою CSS ви можете контролювати широкий спектр візуальних властивостей, включаючи кольори, шрифти, інтервали, вирівнювання тощо. Це дозволяє точно

контролювати зовнішній вигляд веб-сторінок.

2. Контроль макета: CSS надає такі функції, як Flexbox та Grid, які дають розробникам можливість створювати складні макети, адаптовані до різних розмірів екранів.
3. Анімація та переходи: CSS можна використовувати для додавання інтерактивних елементів на веб-сторінку з анімацією та переходами, покращуючи взаємодію з користувачем.
4. Розділення обов'язків: Відокремлення контенту (HTML) від презентації (CSS) спрощує підтримку та оновлення дизайну без впливу на структуру контенту.
5. Зовнішні таблиці стилів: CSS дозволяє використовувати зовнішні таблиці стилів, які можна пов'язати з кількома HTML-документами. Це означає, що ви можете змінити вигляд усього веб-сайту, редагуючи один CSS-файл.

Переваги CSS.

1. Узгодженість: CSS забезпечує однакове стилізування на кількох сторінках, забезпечуючи узгоджений зовнішній вигляд.
2. Ефективність: це дозволяє використовувати менше коду, оскільки стилі можна визначити один раз і застосувати до кількох елементів або сторінок.
3. Підтримка: CSS спрощує підтримку та оновлення дизайну веб-сайту. Зміни в таблиці стилів відображаються на всіх пов'язаних сторінках.
4. Адаптивний дизайн: Медіа-запити в CSS дозволяють створювати адаптивні веб-сайти, які працюють на будь-якому пристрої.
5. Досвід користувача: Покращує взаємодію з користувачем завдяки кращому макету, естетиці та інтерактивним елементам, таким як анімація.

Обмеження CSS.

1. Сумісність браузерів: Різні браузери можуть інтерпретувати CSS по-

різному, що може призвести до невідповідностей та вимагає ретельного тестування.

2. Крива навчання: Для початківців CSS може бути складним і перевантаженим через його різноманітні властивості та синтаксис.
3. Безпека: У CSS бракує безпеки, що означає, що зміни необхідно ретельно перевіряти на наявність потенційних вразливостей.

Випадки використання CSS.

1. Веб-дизайн: CSS використовується для стилізації HTML-документів, контролюючи все, від макета до анімації.
2. Анімація: використовується для додавання інтерактивних елементів до веб-сторінок, таких як анімовані меню та кнопки.
3. Адаптивні шаблони: Багато CSS-фреймворків допомагають створювати динамічні елементи веб-сторінки, які адаптуються до різних розмірів екрана.
4. Соціальні мережі: CSS стилізує інтерфейс користувача платформ соціальних мереж, покращуючи візуальний досвід.

Для розробки коду будемо використовувати універсальний редактор Notepad++, засобами HTML, CSS, JavaScript.

3.2 Створення web-ресурсу

Послідовність розробки та основні етапи для правильної реалізації сайту такі.

Етап 1. Постановка завдання, опис , концепція сайту включає в себе:

- збір даних про поточний стан сайту , в тому числі маркетингової інформації, даних веб - аналітики, минулих рекламних кампаній та іншої активності на сайті; в результаті - постановка завдань створення сайту;
- збір і аналіз вимог до зовнішнього вигляду і функціональності

сайту;

- формування структури сайту, опис прикладів шляхів по сайту, основних "посадочних" сторінок рекламних кампаній (landing pages) [33].

Етап 2. Прототип сайту, створення технічного завдання.

- створення прототипу сайту: набору пов'язаних між собою схем сторінок (wireframes), що демонструють функціонування та структуру сторінок сайту на статичному наборі даних;
- опис прототипу, складання технічного завдання, в яке також включаються функціональні та технічні вимоги до сайту;

Етап 3. Створення дизайну сайту.

- створення концепції дизайну (візуалізація основної сторінки прототипу сайту, наприклад, головної сторінки) , ітерації з доопрацювання концепції, можливе надання додаткових концепцій;
- обговорення припущень, пов'язаних з абстрактністю концепції дизайну (які елементи змінювані);
- створення інших унікальних сторінок дизайну і визначення припущень до подальших сторінкам , які будуть компонуватися в процесі створення;
- створення змінюваних елементів дизайну, передбачених договором та кошторисом (іконки , банери , ілюстрації тощо.);

Етап 4. Html - верстка дизайну і створення додаткових сторінок.

- розробка " сітки " для кожної сторінки дизайну: " сітка" визначає як переводити макет (графічне зображення) в формат html - які елементи будуть зображеннями , які фонами , які текстами , як будуть масштабуватися ті чи інші елементи при зміні екранного дозволу [33];
- розробка додаткових сторінок (карта сайту , результати пошуку

тощо);

- написання скриптів, якщо передбачено дизайном : виділення посилань, зображень, фрагментів тексту , елементів меню при наведенні курсора , при " кліці " і т.п.; оптимізація зображень;
- оптимізація під основні браузер.

Етап 5. Розробка операційної частини сайту (програмування).

- формалізація завдання - від загального опису структури і функцій перехід до конкретних алгоритмів і структури даних;
- переклад html - верстки у формат " движка " (xsl) ;
- розробка структури бази даних;
- написання скриптів адміністрування (back end);
- створення запису на хостингу , веб- інтерфейсу управління базою даних і вмістом сайту;
- написання скриптів візуалізації даних (front end);
- тестова експлуатація системи в процесі наповнення сайту.

Етап 6. Базове наповнення сайту (опціонально).

- автоматизоване наповнення : перенесення інформації з файлів експорту , з БД минулій версії сайту;
- ручне базове наповнення матеріалами.

Слід задуматись про наповнення сайту контентом.

Правила для якісного контенту [20, 21]:

- текст повинен бути написаний цікаво, привертати увагу і затримувати відвідувача.
- у тексті не повинно бути орфографічних помилок, друкарських недоречностей, використання різних елементів в межах одного тексту (наприклад, різних термінів для одного і того ж поняття), це може заплутати читача;
- текст повинен бути унікальним для пошукових систем; брати чужий текст і ставити на свій сайт, змінивши лише назву

компанії, неправильно. за це ваш сайт може бути вилучено з баз пошукових систем;

- текст необхідно оптимізувати під пошукові системи – додати (органічно вписати) достатньо багато ключових слів і виразів;
- текст повинен бути «читабельним» для відвідувачів сайту; часто власники сайту так прагнуть оптимізувати текст, що жоден відвідувач не може винести більш за одне речення [34];
- текст повинен виконувати задану для нього мету, інформувати, коли потрібно інформувати, переконувати, коли потрібно переконувати, продавати, коли потрібно продавати;
- текст необхідно зручно розміщувати на сторінці, щоб відвідувач не плутався у великому масиві тексту без єдиного виділеного рядка і не шукав довго те зображення, яке описується в тексті.

Розробка інтерфейсу сайту.

Будь-який сайт має бути налагоджений так, щоб відвідувачі себе почували зручно і могли без проблем знайти потрібну їм інформацію.

Визначити, що таке добрий інтерфейс можна так: це методи і рішення, спрямовані на те, щоб забезпечити користувачам досягнення мети з найменшими витратами. Інтерфейс залежить від цілей, які переслідує користувач. Подивившись на інтерфейси сучасних веб-сайтів, можемо сказати, що знайти навіть більш-менш добрий – і то проблематично [20, 21, 35].

Для аналізу інтерфейсів візьмемо Інтернет-магазини. Тут наче усе просто: користувач відвідав сайт, переглянув продукцію і поклав в кошик те, що хоче придбати. Але дуже часто процес на чомусь та і завершується. Користувач не зміг оформити подальшого процесу, або не знайшов потрібного йому товару, такого, як він хотів. І усе через проблеми інтерфейсу.

Хоча насправді в створенні якісного інтерфейсу немає нічого складного і надреального.

Головні правила хорошого інтерфейсу такі.

1. Забути про оригінальність.

Засвоювати оригінальний і незвичайний інтерфейс користувачеві доведеться достатньо довго, та і навряд чи, чи займеться він цим. Він просто піде шукати простіший інтерфейс. Цінуйте час своїх відвідувачів!

Тобто інтерфейс – не місце для вираження своєї оригінальності.

2. Вивчати цілі своїх відвідувачів [21].

Добрий інтерфейс завжди задовольняє потреби користувачів. Тому подумайте, проаналізуйте, з якою метою приходять відвідувачі на Ваш сайт. Якщо Вам не вдається думати в такому напрямі, тоді йдіть в зворотному: створюйте сайт, з розрахунку на певні потреби користувачів.

3. Не ускладнювати.

Створюючи ще одну нову можливість на своєму сайті – сто разів подумайте. Швидше за все, вона буде не потрібна Вашим користувачам. І не забувайте про те, що речей одночасно багатофункціональних і зручних у використанні не існує.

4. Не намагатись вирішити інтерфейсом інші завдання.

Це означає, що Ви не зможете керувати тим, що цікавить користувача і примушувати його читати якийсь певний розділ (наприклад, “Про фірму”), навіть якщо він з’являтиметься на кожній сторінці вгорі – це тільки роздратує відвідувача.

5. Визначитись, який саме ресурс потрібно створити.

Інтерфейси мало відвідуваних сайтів і популярних таки відрізняються: популярні можуть собі дозволити використовувати щось нестандартне (тому що з часом користувачі все одно навчаться цьому). Мало відвідувані мають бути максимально зручними і простими.

6. А також не забувати, що скільки б корисної інформації Ваш сайт не містив, він ніколи не доб’ється успіху, якщо цю інформацію просто неможливо на ньому знайти.

Вибір колірної гами.

Вибір кольорової гами відіграє важливу роль при створенні сайтів. Від правильності вибору залежить як відвідувачі будуть сприймати сайт.

Тому, при створенні сайту дуже важливо враховувати:

- особливості фірмового стилю, на підставі якого розробляється сайт;
- фізіологічні і психологічні особливості сприйняття колірної гамми користувачами.

Кольори не повинні бути дратівливими і не напружувати очі. Лише так людина не відволікатиметься від потрібної йому інформації. Користувачу повинен подобатися сайт, він повинен отримувати від нього задоволення і приємне враження.

Оформлення сайту повинно відповідати його тематиці. Якщо це серйозний корпоративний сайт, то не варто робити його строкатим і застосовувати дуже яскраві кольори, він повинен бути строгим, лаконічним. І навпаки, якщо сайт присвячено організації свят, то тьмяне оформлення навряд чи надихне потенційного клієнта.

Окрім відповідності сайту до тематики, колірна гамма повинна бути побудована так, щоб не порушувати основні закономірності впливу кольорів на психологічну реакцію організму. Вона має відповідати стійким гармонійним поєднанням кольорів [21].

Існують певні закономірності для вибору колірної гами:

- кольори повинні відповідати фірмовому стилю компанії;
- комбінація кольорів має відповідати цільовій аудиторії даного сайту;
- не варто використовувати велику кількість кольорів;
- кольори повинні гармоніювати між собою;
- потрібно забезпечити контраст між фоном і звичайним текстом [21].

Колір в дизайні є доволі суб'єктивною категорією. В одної людини він може збудити одні відчуття, а в іншої – абсолютно протилежні. Іноді сприйняття залежить від особистих переваг, іноді – від культурних традицій оточення. Теорія кольору це давня і глибока наука. Деякі фахівці роблять кар'єру, вивчаючи питання, що пов'язані з впливом кольору на індивідуумів і групи людей. І там дійсно є що досліджувати. Елементарна зміна відтінку або насиченості кольору може спровокувати абсолютно несподівані емоції [6].

В світі дизайнерів однієї з головних помилок при проектуванні дизайну сайту є некоректне використання колірної гамми. Невміло підібрана колірна палітра свідчить про непрофесійність дизайнера.

При підборі колірної гамми необхідно враховувати, що сайт буде переглянуто великою кількістю людей з своїми поглядами і смаками, тому, слід ретельно підбирати колірну палітру [21].

До найбільш поширених кольорів у веб дизайні відносяться [36-38]:

- синій (блакитний) - один з найпопулярніших кольорів, що використовується в інтернеті. відноситься до категорії так званих «безпечних» кольорів, який подобається більшості відвідувачів. синій колір асоціюється з такими поняттями, як мир, спокій, надійність, довіра, чесність, чистота, ясність. проте, ясно-блакитний колір може символізувати депресію;
- зелений - завжди асоціюється з природою, символізує оточення, здоров'я, долю, молодість, в деяких випадках може асоціюватися з недовір'ям і небезпекою;
- жовтий - колір радості, щастя, сонця, дружби. серед негативних асоціацій можна виділити боязкість і хворобу, не варто використовувати дуже темний жовтий колір;
- оранжевий - символізує теплоту і енергію, хороший колір для акцентування уваги.

Отже для розробки WEB-ресурсу служби прокату автомобілів обрано аналогічні кольори, основний золотистий колір і суміжні з ним, зелений і оранжевий [36-38].

Розробка динамічного контенту.

Динамічна веб-сторінка (dynamic web page) — веб-сторінка, вміст якої може змінюватись [34].

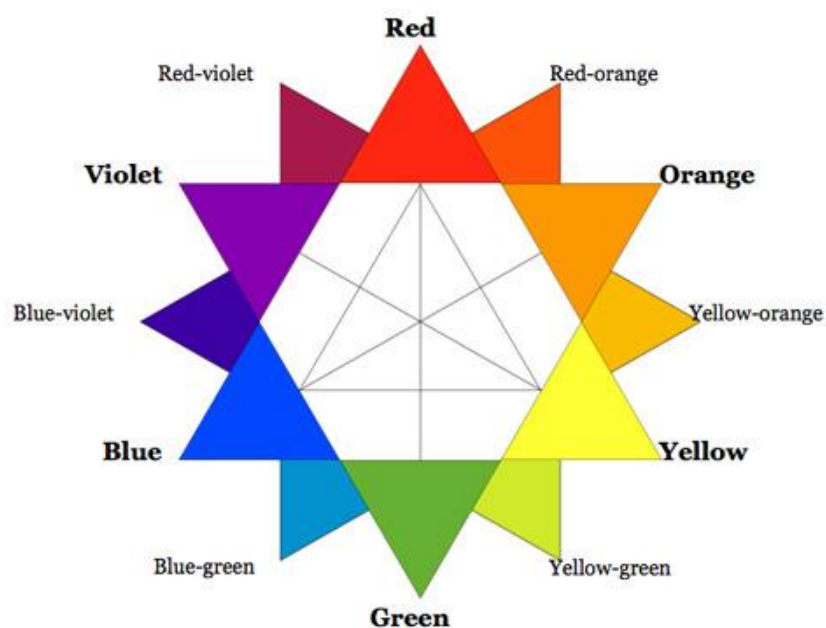


Рисунок 3.1 – Колірна палітра



Рисунок 3.2 – Аналогічні кольори

В початковому варіанті гіпертекстова навігація відбувалася між «статичними» документами. Однак з часом до веб-сторінок було додано інтерактивності, і такі сторінки почали називати динамічними. Наповнення (контент) такої веб-сторінки може замінюватися в залежності від певних умов та/або дій [20-22].

Динамічний контент на сайті - палиця на два кінці. З одного боку - що може бути краще постійних новин, які своєю свіжістю завжди залучають відвідувачів? Але, якщо підійти до постійної зміни контенту з іншого боку, то може виникнути питання - а чи не буде це негативно відбиватися на ранкінгу сторінки в пошукачі?

У принципі, мінуси для сторінки можуть виникнути тільки через те, якщо вона ВСЯ, й причому РАПТОМ, видозміниться. Тому, у цьому випадку, необхідно загострити свою увагу на кількості змінюваного контенту.

Проблеми не будуть виникати, якщо контент змінюється в тих розділах сторінки, які спеціально призначені для таких відновлень, ну а якщо мова про динамічно обновлювані новостні сайти, то питання взагалі відпадає само по собі. Ще приклад - блог. Їжаку зрозуміло, що вміст заголовної сторінки міняється залежно від моторності й переваг блогера.

У наведених прикладах динамічний контент - це двигун прогресу й без нього ніяк. Єдине, що повинен урахувувати веб-майстер, так це елементи сторінки, які залишаються незмінними. Мова йде про заголовок сторінки й про основні елементи навігації - якщо вони статичні, то пошукова система одержує пристойну кількість даних для того, щоб правильно ранжирувати сторінку і видавати її в серп [39].

Незважаючи на вищесказане, зміна контенту може зашкодити репутації сайту, а так само понизити його позиції в пошуковій видачі. Наприклад, якщо взяти сайт, що займає нішу в одному із сегментів електронної комерції, то зміна асортиментів товарів на одній зі сторінок, може негативно позначитися

на всій його діяльності, тому що потенційний клієнт, що прийшов по посиланню з пошукача за конкретним товаром (інформація, про який була проіндексована раніше) і не знайшовши цей товар, буде не зовсім задоволений. Тому, динаміку контенту в подібних вищеописаному випадках, необхідно продумувати завчасно - наприклад, кешувати сторінку або ж задавати якийсь проміжок для її відновлення.

Існує два шляхи для створення динамічних сторінок:

Використання скриптів, що виконуються в браузері користувача (англ. client-side scripting) для зміни вмісту сторінки в залежності від певних дій користувача. Для зміни не потрібно повного перезавантаження сторінки.

Використання програм, що виконуються на сервері (англ. server-side scripting) для зміни наповнення сторінки, що передається браузеру користувача. Інформація може змінюватися в залежності від даних, відправлених у HTML формі, параметрів в URL, типу браузера, дати або часу доби та інших умов.

Результат використання будь-якої техніки може бути описано як динамічну веб-сторінку [38, 39].

Сторінки, побудовані за першим варіантом зазвичай використовують скриптові мови що використовуються для Dynamic HTML (DHTML) — JavaScript або ActionScript. Для додавання відео, звуків та графічних ефектів може бути використано технологію Flash.

Починаючи з 2005 року почала здобувати популярність технологія AJAX, що дозволяє доставляти інформацію з сервера без перезавантаження сторінки.

Сторінки, побудовані за другим варіантом можуть використовувати такі скриптові мови як PHP, Perl, ASP, JSP та інші.

Зміна контенту може зашкодити репутації сайту, а так само понизити його позиції в пошуковій видачі. Наприклад, якщо взяти сайт, що займає нішу в одному із сегментів електронної комерції, то зміна асортиментів

товарів на одній зі сторінок, може негативно позначитися на всій його діяльності, тому що потенційний клієнт, що прийшов по посиланню з пошукача за конкретним товаром (інформація, про який була проіндексована раніше) і не знайшовши цей товар, буде не зовсім задоволений. Тому, динаміку контенту в подібних вищеписаних випадках, необхідно продумувати завчасно.

Для зручності і швидкості пошуку потрібного автомобіля на сайті зроблено динамічний блок який допомагає швидко вибрати категорію автомобілів які цікавлять клієнта. Категорії самостійно перегортаються через деякий час, а також клієнт може самостійно перегортувати їх щоб вибрати ту чи іншу категорію (рис.3.3).



Рисунок 3.3 – Динамічний блок на сайті

На рисунку 3.4 надано вигляд шапки сайту.



Рисунок 3.4 – Шапка сайту

На рисунку 3.5 надано вигляд сторінки каталога автомобілів

 Mazda 323f	Кількість міст:	5	<i>Вартість оренди</i>	
	Об'єм двигуна(л):	2.0	1-4 Дня	55\$
	Коробка передач:	Механіка	5-14 Днів	51\$
	Кондиціонер:	Присутній	15-30 Днів	46\$
	Вид палива:	A-92	30+ Днів	42\$
	Колір:		Застава	700\$
	ЗАМОВИТИ			
 Lancer_X	Кількість міст:	5	<i>Вартість оренди</i>	
	Об'єм двигуна(л):	2.0	1-4 Дня	54\$
	Коробка передач:	Типтронік	5-14 Днів	49\$
	Кондиціонер:	Клімат контроль	15-30 Днів	45\$
	Вид палива:	A-95	30+ Днів	40\$
	Колір:		Застава	600\$
	ЗАМОВИТИ			

Рисунок 3.5 – Каталог автомобілів

На рисунку 3.6 надано вигляд сторінки правил прокату



Рисунок 3.6 – Правила прокату автомобілів

3.3 Тестування web-ресурсу

Під час роботи над сайтом розробник має враховувати багато чинників, які впливають на вид документа. Відвідувачі сайту мають не лише різні операційні системи та браузері, але і такі параметри, як кількість кольорів на моніторі, його роздільність, доступність JavaScript тощо.

Після закінчення верстання слід провести ряд перевірок і у разі виявлення явних помилок, внести в код відповідні зміни. Зрозуміло, це зручніше робити за допомогою спеціалізованих програм.

Відлагодження - це процес знаходження помилок в коді і виправлення небажаного поведінки елементів в браузері. Як правило, відхилення макета від первісного дизайну відстежується в процесі верстання, але бувають ситуації, коли помилки необхідно виправити вже на робочому сайті.

Наприклад, помилка може бути виявлена після додавання нового блоку контенту, тестування сайту в різних версіях браузерів, при різних роздільностях монітора і інших умов. Також розробник повинен вміти швидко розуміти чужий код, відстежити причину появи помилки і усунути її. Розуміння логіки чужого коду потрібне при роботі в команді, або при

поверненні до власної роботи через якийсь час, коли вона вже сприймається як чужа [30, 34].

На об'ємних сайтах з десятками тисяч рядків вихідного коду HTML виділити проблемне місце досить складно, тому потрібно мати інструмент, який дозволяє показати код HTML і CSS вибраного фрагмента і провести над ним експерименти [30].

Firebug один з кращих плагінів під Firefox для веб-розробника. З'явившись в 2007 році, це розширення задало певну планку якості для творців браузерів, що згодом призвело до появи нових версій браузерів, в яких були додані інструменти, за своїм принципом схожі на Firebug.

Більшість існуючих в Інтернеті сторінок, на жаль, не є валідними. Багато розробників вважають, що абсолютно суворе дотримання стандартів HTML - зовсім не обов'язкова умова того, щоб сайт вийшов якісним і ефективним.

Дійсно, існує багато успішних сайтів, код яких не проходить прискіпливої перевірки валідатором [40].

Немає сумнівів, що можна зробити хороший сайт, не домагаючись повної відповідності коду до формальних правил мови. Однак, розробка валідного коду має ряд переваг як для самого розробника, так і для кінцевого користувача. Тому, варто привчити себе писати грамотний код і перевіряти його валідатором.

1. Перевірка документів валідатором дозволяє уникнути дрібних прикрих помилок - неправильно вкладених тегів, пропущених дужок і лапок тощо.
2. Сучасні браузери підтримують стандарти W3C значно краще, ніж їх попередні версії. Ця тенденція зберігається, тому відповідність DTD набуває все більшої важливості для правильного відображення сторінок в браузерах.
3. Оскільки валідний код відповідає певним формальним правилам,

його легше інтерпретувати й обробляти. Він швидше аналізується і відображається в браузері, з ним легше працювати пошуковикам і системам індексації.

4. Валідність коду гарантує сумісність сторінок не лише з існуючими, але і з майбутніми версіями браузерів. Тому, не доведеться переписувати ваші сторінки після виходу нової версії Internet Explorer або Opera.

Валідатор Консорціуму W3C - кращий з існуючих. З його допомогою можна перевірити будь-який документ, що розташований в Інтернеті або на локальному комп'ютері.

Не рекомендується користуватися іншими валідаторами - принаймні до тих пір, поки розробник не навчиться відрізняти справжні валідатори від програм, які просто так називаються.

Піклуючись про грамотність сторінок, про їх відповідність стандартам W3C, не треба впадати в крайнощі і думати, що якщо сторінка успішно перевірена валідатором, то вона автоматично є якісною і ефективною.

Грамотний код - це важливий, але далеко не єдиний показник якості сторінки. Дійсно, можна написати текст, який буде цілком задовольняти граматичним правилам, але при цьому він виявиться невдалим або зовсім безглуздим.

Для перевірки веб-сторінок на наявність помилок і зауважень існує багато підходів. Умовно вони поділяються на онлайнові і локальні. Онлайнові призначені для перевірки сторінок за допомогою браузера через Інтернет, а локальні використовуються для перевірки документів на комп'ютері.

Якщо не приділяти уваги до кросбраузерності, можна втратити багато відвідувачів. Адже перша думка про сайт, складається при перегляді зовнішнього вигляду сайту, його фону та впорядкованого розташування елементів. Некоректне відображення певних елементів дизайну, розбіжність

стиків, може викликати негативну реакцію користувачів, і вони схильні покинути сайт. Пізніше, вони будуть просто ігнорувати цей сайт.

Кросбраузерність - це властивість певного сайту практично однаково відображатися і працювати в браузерах [40]. Сайт має відображати матеріал з однаковим рівнем читабельності, динамічні елементи виконувати всі дії, що закладені розробником. Кросбраузерність є обов'язковою для кожного сайту.

На сьогоднішній день більше 95% користувачів застосовують браузерами Opera, Mozilla Firefox, Google Chrome і Internet Explorer. Не можна з впевненістю сказати, що кожна програма відтворює HTML код або компілює CSS по-своєму, бо вони мають дотримуються певних правил. Але іноді, розробник щось упускає і браузер не відображає властивості елементів або, ж відображає їх некоректно.

Тестувати власні сторінки і сайти слід на стабільних версіях браузера. Таким чином, можна уникнути помилок, які пізніше можуть з'явитися і сприяти збою або неправильному результату. Зазвичай, помилки виникають через недоробки, які були допущені розробниками в програмі в її кодах, або в дизайні. Також, збої можуть виникнути із-за некоректної роботи компілятора, який, відповідно, виробляє некоректний або неправильний код. Отже, щоб уникнути таких прикрощів, потрібно ретельно перевіряти кожен версію браузера [40].

Browsershots - найбільш повний безкоштовний інструмент для тестування. Він містить варіанти браузерів для Linux, Windows і BSD. На ньому можна знайти маловідомі браузери, наприклад, Galeon, Iceape, Kazehakase і Eriphany. За замовчуванням Browsershots проводить перевірку на останніх версіях браузерів, але доступні і попередні. Чим більше браузерів буде визначено для перевірки, тим більше часу піде на підготовку результатів. Даний факт слід враховувати при визначенні набору браузерів.

В ході розробки сайт був протестований на найбільш відомих браузерях Opera, Mozilla Firefox, Google Chrome і Internet Explorer, для яких не було виявлено помилок і некоректного відображення.

3.4 Висновок до розділу 3

У результаті досліджень та розробки виконано програмну реалізація WEB-ресурсу служби прокату автомобілів, обґрунтовано вибір мови програмування, створено web-ресурс, проведено успішне тестування web-ресурсу, яке підтвердило правильність та роботоспроможність розробки.

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 служби прокату автомобілів;
- створено програму WEB-ресурсу служби прокату автомобілів;
- 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 служби прокату автомобілів. Відповідно до завдання було проведено аналіз проблеми та обґрунтування необхідності даної розробки. Розроблено структуру та систему навігації сайту. Наступним етапом проектування було створення інформаційного забезпечення, одного із важливих етапів, який відповідає за наявність на сайті якісного контенту, що призводить до популярності сторінки серед користувачів. Відповідно до всіх 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. Відповідно з завданням до бакалаврської роботи був розроблений WEB-ресурс служби прокату автомобілів. Також було проведено тестування, яке підтвердило роботоспроможність WEB-ресурсу.

Мета роботи - розширення функціональних можливостей WEB-ресурсу служби прокату автомобілів за рахунок розширення каталогу автомобілів і використання дружнього та комфортного інтерфейсу досягнута.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. А.Ю. Подолян, В.О. Денисюк. Розробка WEB-ресурсу служби прокату автомобілів. Молодь в науці: дослідження, проблеми, перспективи (МН-2025). Вінницький національний технічний університет. URL: <https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/viewFile/24656/20405> .
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. Розробка сайту з оренди автомобілів. URL: <https://stfalcon.com/uk/blog/post/car-rent-website-development> .
4. How to make a car rental website to steer your business forward. URL: <https://www.wix.com/blog/how-to-make-a-car-rental-website> .
5. How to Create a Car Rental Website: All You Need to Know. URL: <https://www.codica.com/blog/how-to-build-a-car-rental-website/>
6. Оренда автомобілів: шукайте, порівнюйте і зберігайте. <https://www.rentalcars.com/>
7. Find and compare cheap car rental deals. <https://www.economybookings.com/en> .
8. Uber Rentals is yours for hours. <https://www.uber.com/in/en/ride/rentals/> .
9. Introducing Lyft Rentals. <https://www.lyft.com/blog/posts/introducing-lyft-rentals> .
10. Car rental in Europe, Asia, and Africa. <https://localrent.com/en/> .
11. NarsCars. URL: <https://narscars.com.ua/ua> .
12. Avis. URL: <https://www.avis.com/en/home> .
13. Uklon. URL: <https://uklon.com.ua/> .

14. Rental.ua. URL: <https://rental.ua/ua/vinnitsa> .
15. 7Cars. URL: <https://7cars.com.ua> .
16. BLS Car Rental. URL: <https://bls.ua/ua> .
17. Car-Rent.ua. URL: <https://car-rent.ua> .
18. AutoProkat.net.ua. URL: <https://autoprokat.net.ua> .
19. Payless Car Rental. URL: <https://payless.com.ua> .
20. Романюк О.Н. Основи WEB-дизайну. URL: https://iq.vntu.edu.ua/method/getfile.php?fname=146471.doc&x=1&card_id=71888&id=146471 .
21. Пасічник О.В., Пасічник В.В. Веб-дизайн: Підручник. - 2-ге вид., стер. Львів: ПП “Магнолія 2006”, 2024. 520 с.
22. Етапи створення веб сайтів. URL: <https://webtune.com.ua/statti/web-rozrobka/etapy-stvorenniya-veb-sajtiv/> .
23. 22 Best Languages For Web Development With Its Features [2025]. <https://www.lambdatest.com/blog/best-languages-for-web-development/> .
24. 13 Best Languages for Web Development in 2025. <https://www.browserstack.com/guide/best-language-for-web-development> .
25. Довідник по HTML тегам. URL: <https://css.in.ua/html/tags> .
26. HTML. URL: <https://uk.wikipedia.org/wiki/HTML>
27. Jon Duckett. HTML & CSS: Design and Build Websites. URL: <https://www.htmlandcssbook.com> .
28. Jon Duckett. JavaScript and the DOM. URL: <https://www.javascriptbook.com>.
29. JavaScript. URL: <http://uk.wikipedia.org/wiki/JavaScript>
30. Robin Nixon. Learning PHP, MySQL & JavaScript. O'Reilly. 2018. 799. URL: [https://unidel.edu.ng/focelibrary/books/Learning%20PHP,%20MySQL%20%20JavaScript%20-%20With%20jQuery,%20CSS%20%20HTML5%20by%20Robin%20Nixon%20\(z-lib.org\).pdf](https://unidel.edu.ng/focelibrary/books/Learning%20PHP,%20MySQL%20%20JavaScript%20-%20With%20jQuery,%20CSS%20%20HTML5%20by%20Robin%20Nixon%20(z-lib.org).pdf)
31. CSS. URL: <https://uk.wikipedia.org/wiki/CSS>

32. CSS Tricks – A Complete Guide to Flexbox and Grid. URL: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/> .
33. Технології створення сайтів. URL: <https://dystosvita.org.ua/mod/page/view.php?id=1244> .
34. Динамічна веб-сторінка. URL: http://uk.wikipedia.org/wiki/Динамічна_веб-сторінка.
35. Основні етапи створення сайту. URL: <http://repair.lviv.ua/stvorennnya-sajtiv/osnovni-etapi-stvorennnya-sajtu/> .
36. Психологія кольорів у веб-дизайні. URL: <https://genius.space/lab/psihologiya-koloriv-u-veb-dizajni-yak-vibrati-kolirnu-palitr-shhob-pidvishhiti-konversiyu-sajtiv/>
37. 40 кольорових схем веб-сайтів. URL: <https://www.dreamhost.com/blog/uk/kolirni-skhem-vebsaitu/>
38. Лекції з дисципліни "Веб-дизайн". URL: <https://web-design.okis.ru/leksiia-10-grafichnii-dizain> .
39. Мосіюк О. О. WEB-технології. Частина 1. Верстка. Житомир: Вид-во ЖДУ ім. Івана Франка, 2020. 56 с. URL: http://eprints.zu.edu.ua/32361/1/Web_ost.pdf .
40. 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 служби прокату автомобілівТип 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 24,86 %.

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)

Подольян А.Ю.

(прізвище, ініціали)

ДОДАТОК Б (ОБОВ'ЯЗКОВИЙ)

ЛІСТИНГ ПРОГРАМИ

Лістинг головної сторінки сайту

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>Авто прокат</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />
<link href="style.css" rel="stylesheet" type="text/css" />
<!--<script src="js/jquery-1.9.1.min.js" type="text/javascript"></script>-->
<script src="js/cufon-yui.js" type="text/javascript"></script>
<script src="js/cufon-replace.js" type="text/javascript"></script>
<script src="js/Myrzik_400.font.js" type="text/javascript"></script>

<!--[if lt IE 7]>
  <script type="text/javascript" src="js/ie_png.js"></script>
  <script type="text/javascript">
    ie_png.fix('.png, .list1 li img, #contacts-form input, #contacts-form textarea');
  </script>
<![endif]-->
</head>
<body id="page1">

<div id="header">
  <div class="logo"><a href="index.html"></a></div>
  <div class="site-nav">
    <ul>
      <li><a href="#">Головна</a></li>
      <li><a href="#">Каталог</a></li>
      <li><a href="pravila.html">Правила прокату</a></li>
      <li><a href="kontaktu.html">Контакти</a></li>
    </ul>
  </div>
</div>
```

```

<div id="content">
  <div class="top">
    <div class="bot">
      <div class="inner">
        <div id="forma">
          <div id="img-forma" style="background: url('avto/mazda-323f-ba.jpg'); width: 290px;
height: 206px;"></div>
          <div id="line"><p><b>Mazda 323f</b></p></div>
          <table CELLSPACING="7";>
            <tr>
              <th width="210px" align=left>Кількість мість:</th>
              <td width="140px"; align=right>5</td>
              <td width="95px"; align=right></td>
              <td width="60px"; align=right></td>
            </tr>
            <tr>
              <th align=left>Об'єм двигуна(л):</th>
              <td align=right>2.0</td>
              <td align=right>1-4 Дня</td>
              <td align=right>55$</td>
            </tr>
            <tr>
              <th align=left>Коробка передач:</th>
              <td align=right>Механіка</td>
              <td align=right>5-14 Днів</td>
              <td align=right>51$</td>
            </tr>
            <tr>
              <th align=left>Кондиціонер:</th>
              <td align=right>Присутній</td>
              <td align=right>15-30 Днів</td>
              <td align=right>46$</td>
            </tr>
            <tr>
              <th align=left>Вид палива:</th>
              <td align=right>А-92</td>
              <td align=right>30+ Днів</td>

```

```

<td align=right>42$</td>
</tr>
<tr>
<th align=left>Колір:</th>
<td align=right><div id="color_avto" style="background:#ffffff"></div>
</td>

<td align=right>Засрава</td>
<td align=right>700$</td>
</tr>
</table>

</div>
<div id="button">
<h2>
<a href="#">заповіти</a>
</h2>
</div>
<br>
<!------->
<div id="forma">
<div id="img-forma" style="background: url('avto/lancer_x.jpg'); width: 290px; height: 206px;">
</div>
<div id="line"><p><b>Lancer_x</b></p></div>
<table CELLSPACING="7">
<tr>
<th width="210px" align=left>Кількість місць:</th>
<td width="140px"; align=right>5</td>
<td width="95px"; align=right></td>
<td width="60px"; align=right></td>
</tr>
<tr>
<th align=left>Об'єм двигуна(л):</th>
<td align=right>2.0</td>
<td align=right>1-4 Дня</td>
<td align=right>54$</td>
</tr>

```

```

<tr>
<th align=left>Коробка передач:</th>
<td align=right>Типтроник</td>
<td align=right>5-14 Днів</td>
<td align=right>49$</td>
</tr>
<tr>
<th align=left>Кондиціонер:</th>
<td align=right>Клімат контроль</td>
<td align=right>15-30 Днів</td>
<td align=right>45$</td>
</tr>
<tr>
<th align=left>Вид палива:</th>
<td align=right>А-95</td>
<td align=right>30+ Днів</td>
<td align=right>40$</td>
</tr>
<tr>
<th align=left>Колір:</th>
<td align=right><div id="color_avto" style="background:#700e10"></div></td>
<td align=right>Застава</td>
<td align=right>600$</td>
</tr>
</table>

</div>
<div id="button">
<h2>
<a href="#">заказати</a>
</h2>
</div>
<br>
<!------->
<div id="forma"><div id="img-forma" style="background: url('avto/daewoo_lanos.jpg');
width: 290px;

```

```

height: 206px;"></div>
<div id="line"><p><b>Daewoo Lanos</b></p></div>
    <table CELLSPACING="7">
        <tr>
            <th width="210px" align=left>Кількість мість:</th>
            <td width="140px"; align=right>5</td>
            <td width="95px"; align=right></td>
            <td width="60px"; align=right></td>
        </tr>
        <tr>
            <th align=left>Об'єм двигуна(л):</th>
            <td align=right>1,5</td>
            <td align=right>1-4 Дня</td>
            <td align=right>35$</td>
        </tr>
        <tr>
            <th align=left>Коробка передач:</th>
            <td align=right>Механіка</td>
            <td align=right>5-14 Днів</td>
            <td align=right>32$</td>
        </tr>
        <tr>
            <th align=left>Кондиціонер:</th>
            <td align=right>Присутній</td>
            <td align=right>15-30 Днів</td>
            <td align=right>27$</td>
        </tr>
        <tr>
            <th align=left>Вид палива:</th>
            <td align=right>А-95</td>
            <td align=right>30+ Днів</td>
            <td align=right>22$</td>
        </tr>
        <tr>
            <th align=left>Колір:</th>
            <td align=right><div id="color_avto" style="background:#DCDCDC">

```

```

        <td align=right>Застава</td>
        <td align=right>400$</td>
    </tr>
</table>

</div>
<div id="button">
    <h2>
<a href="#">замавити</a>
    </h2>
</div>
<br>
<!------->
    <div id="forma">
<div id="img-forma" style="background: url('avto/almera.jpg'); width: 290px; height: 206px;">
</div>
    <div id="line"><p><b>Nissan Almera</b></p></div>
    <table CELLSPACING="7">
        <tr>
            <th width="210px" align=left>Кількість місць:</th>
            <td width="140px"; align=right>5</td>
            <td width="95px"; align=right></td>
            <td width="60px"; align=right></td>
        </tr>
        <tr>
            <th align=left>Об'єм двигуна(л):</th>
            <td align=right>1,6</td>
            <td align=right>1-4 Дня</td>
            <td align=right>45$</td>
        </tr>
        <tr>
            <th align=left>Коробка передач:</th>
            <td align=right>Автомат</td>
            <td align=right>5-14 Днів</td>
            <td align=right>40$</td>
        </tr>
    </table>

```



```

<tr>
<th align=left>Кондиціонер:</th>
<td align=right>Присутній</td>
<td align=right>15-30 Днів</td>
<td align=right>36$</td>
</tr>
<tr>
<th align=left>Вид палива:</th>
<td align=right>А-95</td>
<td align=right>30+ Днів</td>
<td align=right>33$</td>
</tr>
<tr>
<th align=left>Колір:</th>
<td align=right><div id="color_avto" style="background:#969594"
></div></td>
<td align=right>Застава</td>
<td align=right>600$</td>
</tr>
</table>

</div>
<div id="button">
<h2>
<a href="#">замовити</a>
</h2>
</div>
<br>

<!------->
<div id="forma">
<div id="img-forma" style="background: url('avto/honda_civic.jpg'); width: 290px; height: 206px;">
</div>
<div id="line"><p><b>Honda Civic</b></p></div>
<table CELLSPACING="7";>
<tr>

```

```

<th width="210px" align=left>Кількість місць:</th>
<td width="140px"; align=right>5</td>
<td width="95px"; align=right></td>
<td width="60px"; align=right></td>
</tr>
<tr>
<th align=left>Об'єм двигуна(л):</th>
<td align=right>1.8</td>
<td align=right>1-4 Дня</td>
<td align=right>55$</td>
</tr>
<tr>
<th align=left>Коробка передач:</th>
<td align=right>Автомат</td>
<td align=right>5-14 Днів</td>
<td align=right>51$</td>
</tr>
<tr>
<th align=left>Кондиціонер:</th>
<td align=right>Клімат контроль</td>
<td align=right>15-30 Днів</td>
<td align=right>46$</td>
</tr>
<tr>
<th align=left>Вид палива:</th>
<td align=right>А-95</td>
<td align=right>30+ Днів</td>
<td align=right>42$</td>
</tr>
<tr>
<th align=left>Колір:</th>
<td align=right><div id="color_avto" style="background:#1d5496">
</div></td>
<td align=right>Застава</td>
<td align=right>600$</td>
</tr>
</table>

```

```
</div>
<div id="button">
  <h2>
    <a href="#">заповіти</a>
  </h2>
</div>
<br>
</div>

  </div>
</div>
<div id="footer">

</div>
```

Файл стилів CSS

```

/* TAGS */
* {margin:0;padding:0}
html {min-width:1000px}
body {background:#000;font-family:Tahoma, Arial, helvetica,
sans-serif;font-size:100%;line-height:1em;color:#000;min-width:1000px}
ul, ol {list-style:none}
/* GLOBAL */
#header, #content, #footer {font-size:0.75em;width:1000px;margin:0 auto}

/* LISTS */
.site-nav {position:absolute;left:36px;top:292px}
.site-nav ul li {float:left;font-size:30px;line-height:1.2em;padding:0 47px}
.site-nav ul li a {text-decoration:none;color:#000}
.site-nav ul li a:hover {color:#6e6774}

/* HEADER */
#header {height:337px;position:relative;background:url(images/header-bg.jpg) no-repeat left top}
#header .logo {position:absolute;left:50px;top:94px}
/* CONTENT */
#content {background:url(images/cont-tail.jpg) repeat-y left 216px}
#content .top {background:url(images/cont-top.jpg) no-repeat left top;}
#content .bot {background:url(images/cont-bot.jpg) no-repeat left bottom;min-height:713px;
height:auto !important;height:713px}
#content .inner {padding:65px 0 0px 50px;}

/* FOOTER */
#footer {height:57px;background:url(images/footer-bg.jpg) no-repeat left top}
/* #footer .indent {padding:5px 129px 20px 94px}
#footer img {position:relative;top:-4px} */
/* */
#forma {
    background:url('avto/forma.png') no-repeat;
    width: 892px;
    height: 234px;
}

```

```
#color_avto {
    width: 20px;
    height: 20px;
}
#line{
    position:relative;
    bottom:45px;
    left:19px;
    width:290px; /* Довжина фігури */
    height:30px; /* Висота фігури */
    background:#ffffff; /* фон блока */
    opacity: 0.6;
}
```

```
#line p {
    position:relative;
    top:6px;
    font-family: Tahoma;
    font-size: 22px;
    font-style: normal;
    text-align: center;
}
```

```
#img-forma {
    border-radius: 10px;
    -webkit-border-radius: 10px;
    -moz-border-radius: 10px;
    position: relative;
    left:19px;
    top:13px;
}
```

```
#button {
    position:relative;
    top:5px;
    left:300px;
    width:300px;
    height:20px;
```

```
        background:#B5B5B5;
    }

#button h2 {
    font-size: 15px;
    color: #3a454e;
    text-align:center;
    color:black;
}

#button a:hover {font-size: 16px;}

table {
    position: relative;
    left:330px;
    bottom:233px;
    font-size: 15px;
}

tr {
    height:30px;
}

#pravila_verch {
    background:url('images/prav.png');
    width:767px;
    height:45px;
    position:relative;
    left:55px;
}

#pravila {    padding-top:20px;
    padding-left:40px;
}

#pravila p {    font-size: 15px;
}
```

ДОДАТОК В (ОБОВ'ЯЗКОВИЙ)

ІЛЮСТРАТИВНА ЧАСТИНА

РОЗРОБКА WEB-РЕСУРСУ СЛУЖБИ ПРОКАТУ
АВТОМОБІЛІВ

Виконав: студент 4-го курсу,
групи ЗКН-226
спеціальності 122 «Комп'ютерні науки»
(цифр і назва напрямку підготовки, спеціальності)

Подольн А.Ю.
(прізвище та ініціали)

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

« 3 » серпня 2025 р.

Вінниця ВНТУ - 2025 рік

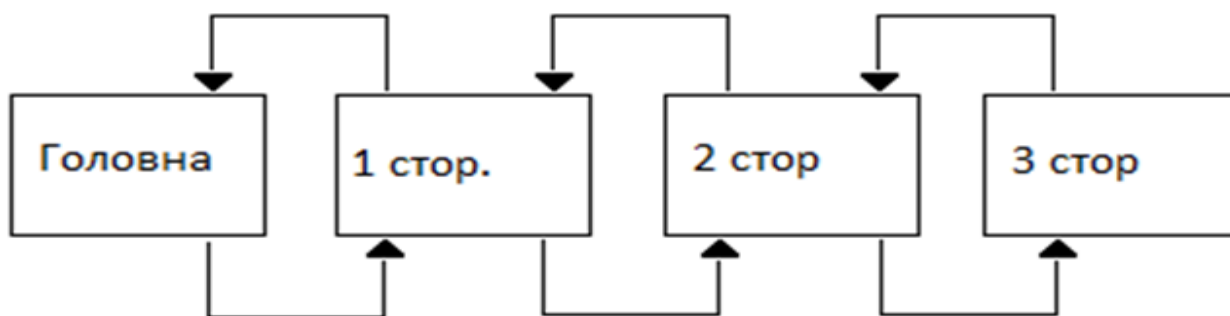


Рисунок В.1 – Каскадна структура сайту

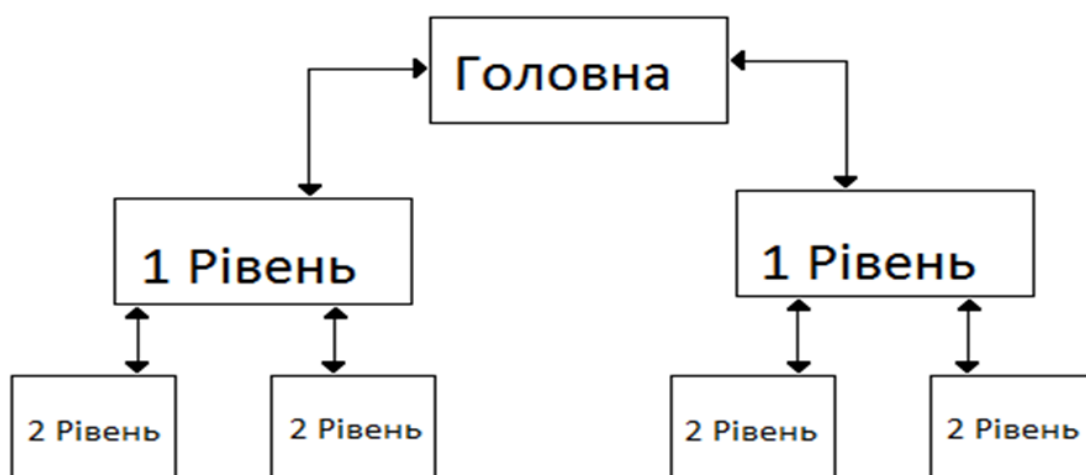


Рисунок В.2 – Деревоподібна структура сайту

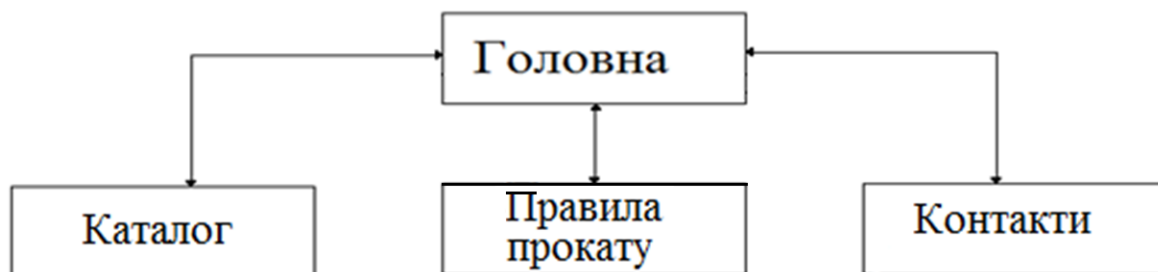


Рисунок В.3 – Вільна структура сайту

Шапка сайту
Меню сайту
Контент
Нижня частина сайту

Рисунок В.4 – Вигляд сторінки сайту

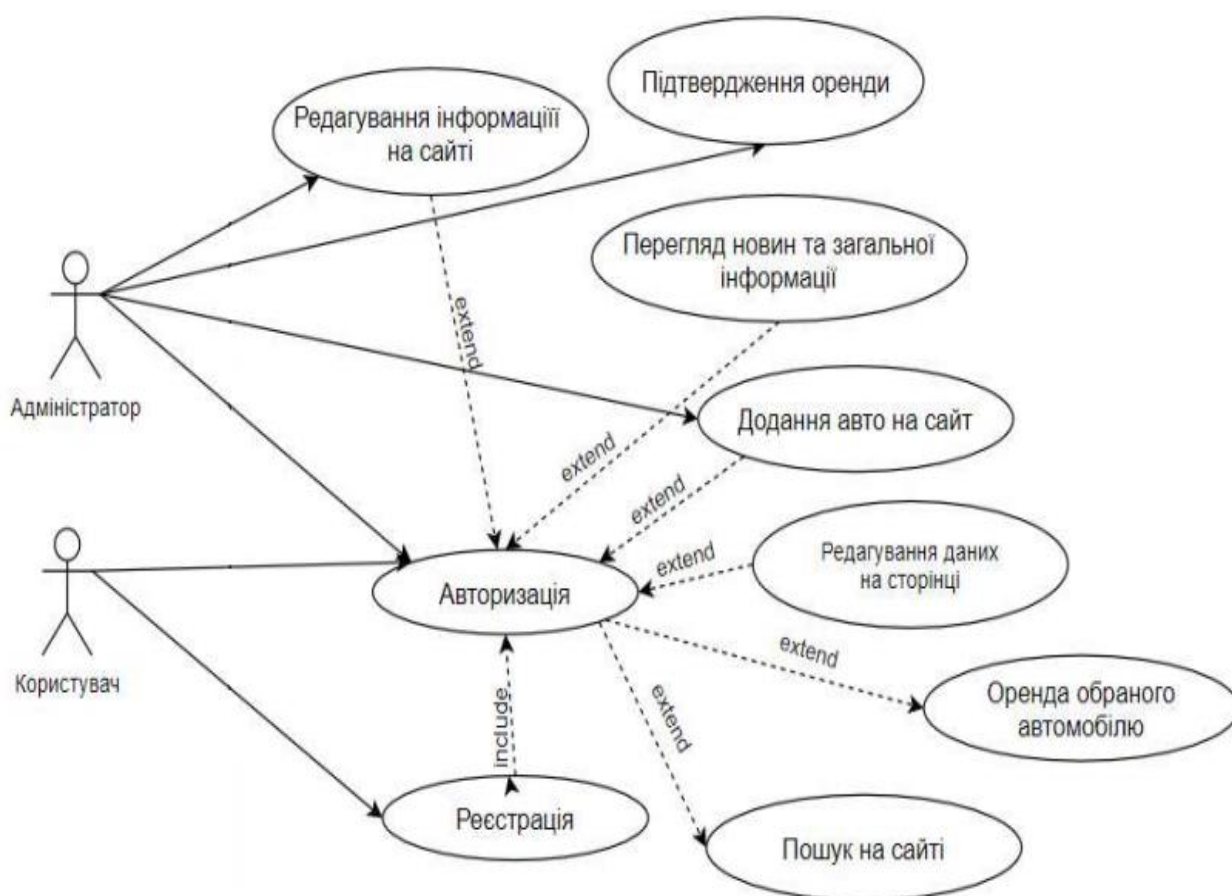


Рисунок В.5 – UML-діаграма варіантів використання

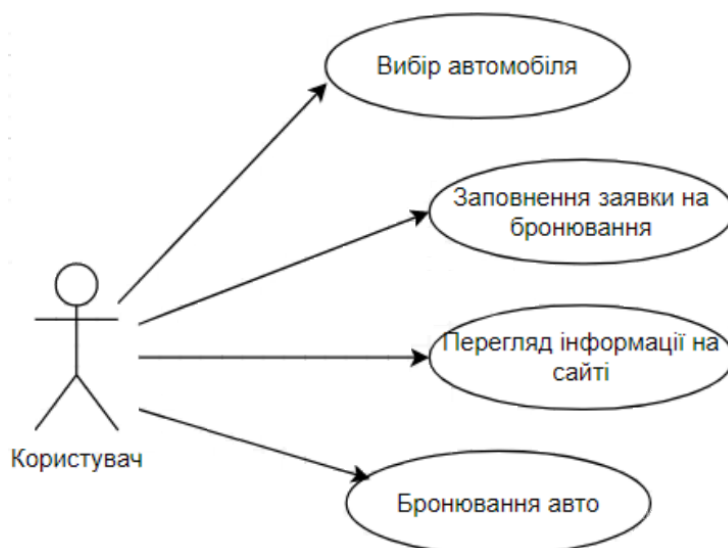


Рисунок В.6 - UML-діаграма прецедентів “Бронювання автомобіля”



Рисунок В.7 - UML-діаграма прецедентів “Авторизація адміністратора”

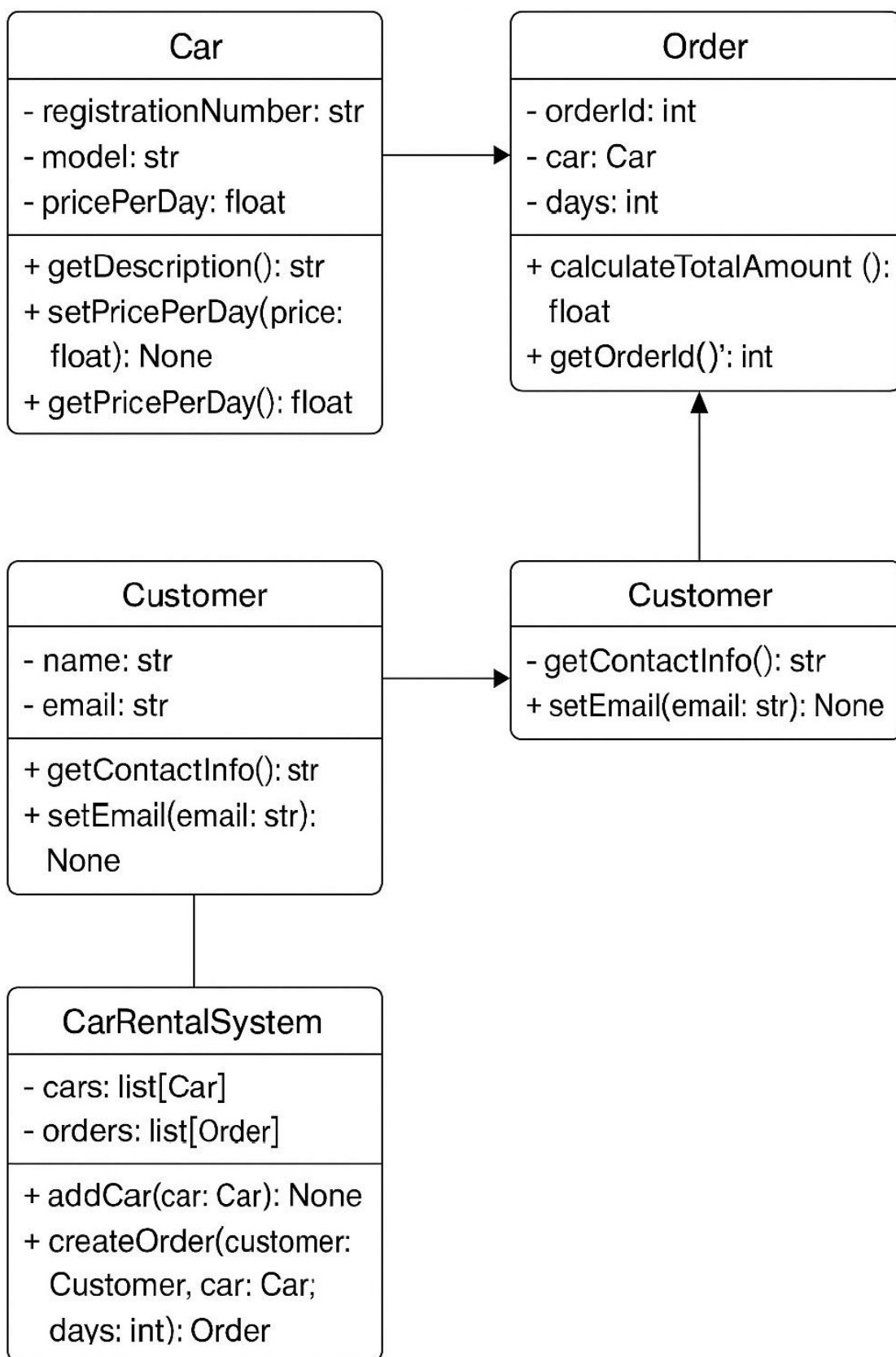


Рисунок В.8 - UML-діаграма класів

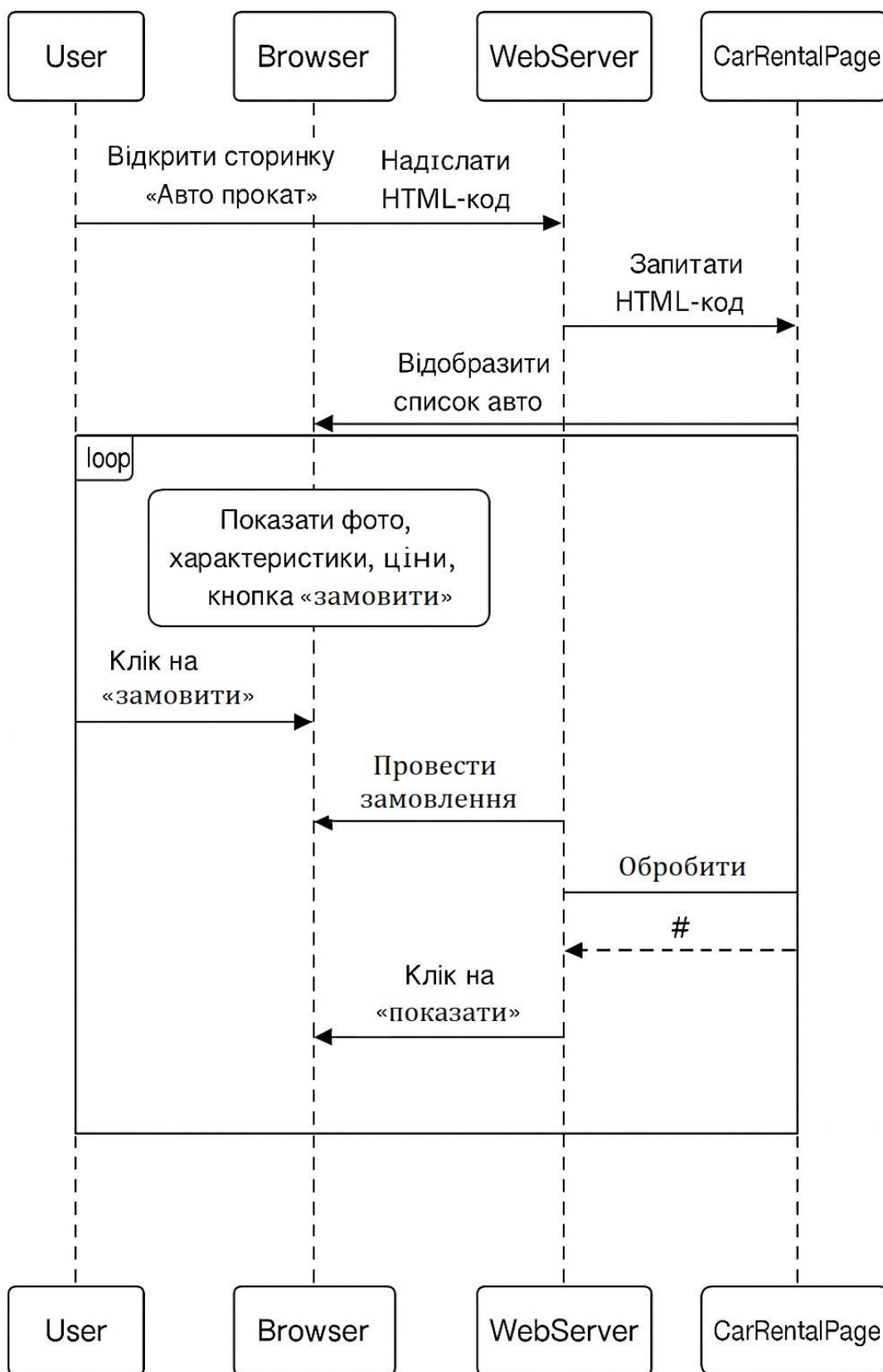


Рисунок В.9 - UML діаграму послідовності дій WEB-ресурсу
служби прокату автомобілів

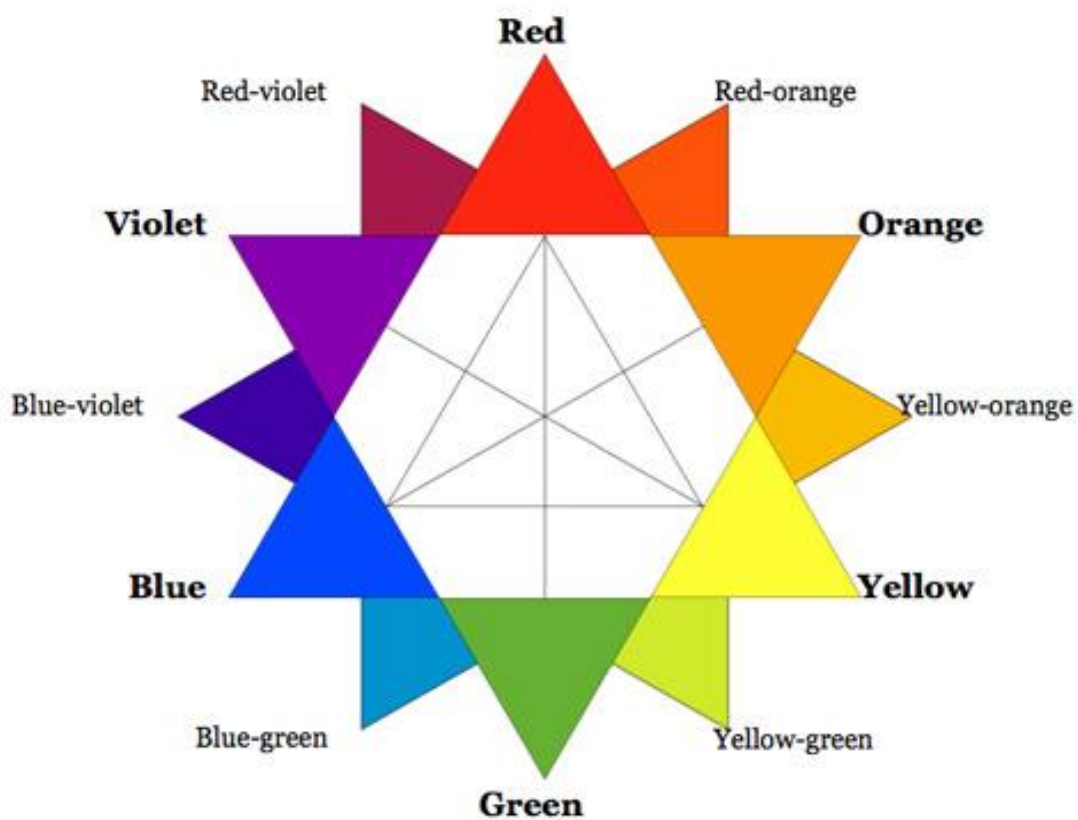


Рисунок В.10 – Колірна палітра



Рисунок В.11 – Аналогічні кольори



Рисунок В.12 – Динамічний блок на сайті



Рисунок В.13 – Шапка сайту

 Mazda 323f	Кількість мість:	5	<i>Вартість оренди</i>	
	Об'єм двигуна(л):	2.0	1-4 Дня	55\$
	Коробка передач:	Механіка	5-14 Днів	51\$
	Кондиціонер:	Присутній	15-30 Днів	46\$
	Вид палива:	A-92	30+ Днів	42\$
	Колір:		Застава	700\$
	ЗАМОВИТИ			
 Lancer_x	Кількість мість:	5	<i>Вартість оренди</i>	
	Об'єм двигуна(л):	2.0	1-4 Дня	54\$
	Коробка передач:	Типтронік	5-14 Днів	49\$
	Кондиціонер:	Клімат контроль	15-30 Днів	45\$
	Вид палива:	A-95	30+ Днів	40\$
	Колір:		Застава	600\$
	ЗАМОВИТИ			

Рисунок В.14 – Каталог автомобілів



Рисунок В.15– Правила прокату автомобілів

ДОДАТОК Г (ДОВІДНИКОВИЙ)

ІНСТРУКЦІЯ КОРИСТУВАЧА

Загальна інформація.

Вебсайт дозволяє переглядати доступні автомобілі для оренди, переглядати технічні характеристики, ціни оренди на різні терміни, а також оформити замовлення на конкретний автомобіль.

Структура сайту.

Головне меню (нагорі сторінки) містить

- **Головна** – повернення на головну сторінку з усіма авто;
- **Каталог** – перегляд усіх доступних автомобілів;
- **Правила прокату** – сторінка з умовами оренди;
- **Контакти** – інформація для зв'язку з компанією.

Перегляд автомобілів

На головній сторінці представлено перелік автомобілів. Кожен блок містить:

- **Фото автомобіля**
- **Назву моделі**
- **Таблицю** з такими характеристиками:
 - Кількість місць
 - Об'єм двигуна
 - Коробка передач
 - Кондиціонер
 - Вид пального
 - Колір авто (візуально показано)
 - Ціни за різні терміни прокату:

- 1–4 дні
- 5–14 днів
- 15–30 днів
- 30+ днів
- Розмір застави

Замовлення.

1. Знайдіть автомобіль, який вас цікавить.
2. Натисніть кнопку «**замовити**» під відповідним авто.
3. (У поточній реалізації кнопка не прив'язана до форми — для реального оформлення необхідно реалізувати форму замовлення або скрипт на сервері.)

Примітка: після натискання кнопки "замовити" може бути реалізована переадресація на сторінку з формою, яка ще не включена у цей шаблон.

Умови оренди.

Щоб переглянути детальні **правила прокату**, натисніть у меню пункт «**Правила прокату**». Там мають бути вказані:

- Необхідні документи
- Мінімальний вік/стаж водія
- Умови застави
- Обмеження по пробігу тощо

Контактна інформація.

Для зв'язку з компанією:

- 1) натисніть пункт меню «**Контакти**»;
- 2) там має бути вказана адреса, номер телефону, email або форма зворотного зв'язку.

Технічні вимоги.

Сайт розроблений на XHTML 1.0 Strict, тому працює у більшості сучасних браузерів. Для правильного відображення необхідно:

- увімкнена підтримка JavaScript;
- шрифти Cufon (підключено через `js/cufon-yui.js`);
- папки `images`, `avto`, `js`, `style.css` мають бути присутні поруч із HTML-файлом.

Важлива інформація:

- всі дані (ціни, застави, характеристики авто) вказані в HTML-файлі вручну — тому вони не оновлюються динамічно;
- кнопки "замовити" **не прив'язані до бази даних** чи форм - потрібна серверна частина або email-форма, якщо планується реальна робота з клієнтами.