

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет

імені Петра Могили

Факультет комп'ютерних наук

Кафедра інтелектуальних інформаційних систем

Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.

_____ Ю. П. Кондратенко

« ____ » _____ 2023 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

ВЕБЗАСТОСУНОК З ПРОДАЖУ АВТОМОБІЛІВ

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 401.21910126

Виконав студент 4-го курсу, групи 401

_____ *Д.В. Шевченко*

« ____ » червня 2023 р.

Керівник: канд. техн. наук, доцент

_____ *Є. В. Сіденко*

« ____ » червня 2023 р.

Миколаїв – 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Рівень вищої освіти **бакалавр**
Спеціальність **122 «Комп'ютерні науки»**
(шифр і назва)
Галузь знань **12 «Інформаційні технології»**
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.
_____ Ю. П. Кондратенко
« ____ » _____ 20__ р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

Видано студенту групи 401 факультету комп'ютерних наук Шевченко Дмитру Віталійовичу.

1. Тема кваліфікаційної роботи «Вебзастосунок з продажу автомобілів».

Керівник роботи Сіденко Євген Вікторович, канд. техн. наук, доцент.

Затв. наказом Ректора ЧНУ ім. Петра Могили від « ____ » ____ 20__ р. № _____

2. Строк представлення кваліфікаційної роботи студентом « ____ » _____ 20__ р.

3. Вхідні (початкові) дані до роботи: оцінки технологій розробки Інтернет-магазинів

Очікуваний результат роботи: зручний та приємний для використання інтернет-магазин з продажу автомобілів.

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):

- аналіз стану поставленої задачі вибору технології для програмної реалізації;
- огляд існуючих методів розробки інтернет-магазинів;
- оцінювання технологій розробки інтернет-магазину за визначеними критеріями;

5. Перелік графічних матеріалів презентація.

6. Завдання до спеціальної частини:

- здійснити аналіз умов праці в робочому приміщенні;
- встановити необхідний рівень показників для робочого приміщення, де проводяться роботи з розробки системи;
- встановити основні принципи техніки безпеки.

7. Консультанти:

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина з охорони праці	Алексєєва А.О., канд. техн. наук, доцент	

Керівник роботи канд. техн. наук, доцент, Сіденко Є.В.
(наук. ступінь, вчене звання, прізвище та ініціали)

(підпис)

Завдання прийнято до виконання Шевченко Д.В.
(прізвище та ініціали)

(підпис)

Дата видачі завдання «__» _____ 202__ р.

КАЛЕНДАРНИЙ ПЛАН виконання кваліфікаційної роботи

Тема: Вебзастосунок з продажу автомобілів.

№	Найменування роботи	Початок	Закінчення	Примітки
1	Подання заяви на затвердження теми та керівників БКР	30.10.2022	30.10.2022	Виконано
2	Отримання завдання на виконання БКР	21.11.2022	21.11.2022	Виконано
3	Складання календарного плану роботи на весь період виконання БКР	10.02.2023	10.02.2023	Виконано
4	Отримання завдання на переддипломну практику	25.04.2023	25.04.2023	Виконано
5	Проходження переддипломної практики, збір та аналіз матеріалів до БКР	01.05.2023	14.05.2023	Виконано
6	Оформлення звіту з переддипломної практики	14.05.2023	14.05.2023	Виконано
7	Виконання БКР: аналіз алгоритмів, огляд існуючих рішень, формування задачі, розробка інформаційної системи	15.02.2023	15.06.2023	Виконано
8	Оформлення звіту БКР	28.04.2023	18.06.2023	Виконано
9	Попередній захист БКР на засіданні комісії кафедри	29.05.2023	29.05.2023	Виконано
10	Доробка та виправлення помилок в звіті	30.05.2023	18.06.2023	Виконано
11	Подання БКР рецензенту	18.06.2023	18.06.2023	Виконано
12	Подання БКР, її електронної копії та інших документів (відгуку, рецензії) до захисту	21.06.2023	21.06.2023	Виконано
13	Захист БКР перед екзаменаційною комісією (ЕК)	28.06.2023	28.06.2023	Виконано

Розробив студент Шевченко Д.В.
(прізвище та ініціали)

_____ (підпис)

Керівник роботи канд. техн. наук, доцент, Сіденко Є.В.
(наук. ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

« _____ » _____ 202__ р.

АНОТАЦІЯ

**бакалаврської кваліфікаційної роботи
студента групи 401 ЧНУ ім. Петра Могили**

Шевченка Дмитра Віталійовича

Тема: «Вебзастосунок з продажу автомобілів»

Актуальність – зростання популярності онлайн-торгівлі значно змінило підхід до придбання товарів, включаючи й автомобілі тому розробка інтернет-магазину, спеціалізованого на продажі автомобілів, стає актуальною темою для роботи.

Об'єкт роботи – процес розробки інтернет-магазину .

Предмет роботи – структура бази даних, додавання, редагування та видалення записів про автомобілі, збереження деталей про характеристики, фотографії, ціни та іншу інформацію.

Метою бакалаврської кваліфікаційної роботи є надання користувачам зручної та доступної платформи для перегляду, вибору та придбання автомобілів. Головним завданням є створення простого та інтуїтивно зрозумілого інтерфейсу, який дозволить покупцям знайти потрібний автомобіль, переглянути детальну інформацію про нього, порівняти різні моделі, а також здійснити покупку онлайн з мінімальними зусиллями

Пояснювальна записка складається зі вступу, чотирьох розділів, висновків та додатків.

У першому розділі проведено опис сфери розробки інтернет-магазинів, проаналізовано аналоги.

У другому розділі досліджено основні методи вирішення поставленої задачі.

У третьому розділі описано створення та налаштування дизайну макету інтернет-магазину та його програмна реалізація.

У четвертому розділі наведено результати розробки інтернет-магазину, під'єднання бази даних, інформаційні діаграми.

В результаті розроблено зручний у використанні інтернет-магазин.

Бакалаврська кваліфікаційна робота містить __ сторінок, __ рисунків, __ таблиць, __ використаних джерел та __ додатків.

ABSTRACT

**Bachelor's qualification work
of the student of 401 group of Petro Mohyla Black Sea National University
Shevchenko Dmytro**

Title: «Web application for car sales»

Relevance - the growing popularity of online trading has significantly changed the approach to purchasing goods, including automobiles. The development of an online store specializing in car sales becomes a relevant topic to explore.

The object of the work is the processes involved in developing an online car marketplace.

The subject of the work is the software tools and technologies used in creating web applications.

The aim of the bachelor's thesis is to develop a web application for car sales that provides users with a convenient and accessible platform for browsing, selecting, and purchasing cars. The explanatory note consists of an introduction, four sections, conclusions, and appendices.

The first section provides a description of the internet store development field and analyzes analogs.

The second section explores the main methods for solving the set task.

The third section describes the creation and configuration of the internet store's layout design and its software implementation.

The fourth section presents the results of the internet store development, database connection, and informational diagrams.

As a result, a user-friendly internet store has been developed.

The bachelor's qualification work consists of __ pages, __ figures, __ tables, __ references, and __ ads.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП.....	4
1 АНАЛІЗ СФЕРИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕРНЕТ-МАГАЗИНІВ	6
1.1 Опис сфери розробки інтернет-магазинів.....	6
1.2 Аналіз аналогів інтернет-магазину з продажу автомобілів	8
1.3 Специфікація вимог до програмного забезпечення інтернет-магазину з продажу автомобілів	12
2 МОДЕЛІ, МЕТОДИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ.....	15
2.1 HTML5 – мова розмітки гіпертексту.....	15
2.2 Таблиця стилів CSS3.....	19
2.3 Мова програмування JavaScript.....	21
2.4 Бази даних.....	24
2.5 PHP.....	26
3 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОГО ВЕБЗАСТОСУНКУ.....	29
3.1 Макет користувацького інтерфейсу	29
Рисунок 3.1– Макет головної сторінки магазину	30
3.2 Візуальнаа карта веб-застосунку	39
3.3 Сценарії використання	41
Висновки до розділу 3	43
4 ТЕСТУВАННЯ РОЗРОБЛЕНОГО ВЕБЗАСТОСУНКУ	44
4.1 Опис інтерфейсу вебзастосунку	44
4.2 Фізична модель бази даних.....	50
4.3 Інструкція для користувача вебзастосунку.....	53
Висновки до розділу 4	55
ВИСНОВКИ	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	59
ДОДАТОК А	61

ПЕРЕЛІК СКОРОЧЕНЬ

БД	–	База даних
СКБД	–	Система керування базами даних
CMS	–	Content Management System
CSS	–	Cascading Style Sheets
HTML	–	HyperText Markup Language
MySQL	–	My Structured Query Language
PHP	–	Personal Home Page (Tools), Hypertext Preprocessor
JS	–	JavaScript

ВСТУП

В сучасному цифровому світі інтернет-магазини стали невід'ємною частиною електронної комерції, забезпечуючи зручну та доступну платформу для купівлі та продажу товарів в онлайн-режимі. За останні роки зростання популярності онлайн-торгівлі значно змінило підхід до придбання товарів, включаючи й автомобілі. Розробка інтернет-магазину, спеціалізованого на продажі автомобілів, стає актуальною і перспективною темою для бакалаврської роботи.

Мета даної роботи полягає в розробці та реалізації інтернет-магазину, який надасть користувачам зручну платформу для перегляду, вибору та придбання автомобілів. Головним завданням цього проекту є створення ефективної та легко використовуваної системи, яка дозволить покупцям отримати детальну інформацію про доступні автомобілі, здійснювати пошук за певними критеріями, здійснювати онлайн-замовлення та оплату, а також забезпечити зручний процес комунікації з продавцями.

У контексті даного дослідження важливо врахувати різноманітні аспекти, пов'язані з розробкою інтернет-магазину для продажу автомобілів. Серед них - вибір технологічних рішень для створення функціональності веб-додатку, проектування інтерфейсу, розробка бази даних для зберігання інформації про автомобілі, розробка системи пошуку та фільтрації, впровадження механізмів онлайн-оплати та забезпечення безпеки та конфіденційності даних

Хотілося б відокремити основні завдання даної роботи:

- визначення основних функцій інтернет-магазину;
- створення приємного та ручного дизайну;
- аналіз конкурентного середовища;
- розробка прототипу інтернет- магазину.

Результати проведених досліджень та розробок інтернет-магазину мають значну практичну цінність для сфери електронної комерції. Ця робота допоможе краще зрозуміти принципи розробки та удосконалення інтернет-магазинів, а також

надасть рекомендації для покращення ефективності торгової діяльності через онлайн-канали.

1 АНАЛІЗ СФЕРИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕРНЕТ-МАГАЗИНІВ

1.1 Опис сфери розробки інтернет-магазинів

Предметною сферою розробки інтернет-магазинів є електронна комерція, яка є надзвичайно важливою для сучасного бізнесу. Електронна комерція включає в себе купівлю та продаж товарів і послуг через Інтернет-магазини, що дозволяє клієнтам замовляти товари зручним для них способом та отримувати їх у відповідності до власних потреб та побажань.

Розробка інтернет-магазину вимагає розуміння основних принципів та інструментів електронної комерції, таких як онлайн-маркетинг, електронний платіж, системи управління взаємодією з клієнтами, обробка замовлень, доставка товарів, аналітика та звітність.

У зв'язку зі зростаючою популярністю електронної комерції, розробка інтернет-магазину стає все більш вимогливою та складною задачею. Найважливіші аспекти, які потрібно враховувати при розробці інтернет-магазину:

- зручність користувача: інтерфейс магазину повинен бути інтуїтивно зрозумілим та легким у використанні для клієнтів. Навігація має бути зрозумілою, пошукова система - ефективною, а процес замовлення - простим та швидким;
- дизайн та брендінг: важливо створити привабливий та професійний дизайн магазину, який відповідає бренду компанії. Це створює довіру серед клієнтів і сприяє позитивному враженню про товари та послуги;
- мобільна сумісність: з урахуванням зростаючої кількості користувачів, які здійснюють покупки з мобільних пристроїв, важливо забезпечити адаптивний дизайн, що забезпечить коректне відображення та зручне використання магазину на різних пристроях;

- безпека: захист особистих даних клієнтів та безпека оплати є надзвичайно важливими. Варто забезпечити застосування шифрування даних, використання безпечних протоколів передачі інформації та надійну систему оплати;
- швидкість та продуктивність: інтернет-магазин повинен працювати швидко і ефективно, з мінімальними затримками завантаження сторінок. Це важливо для забезпечення задоволення клієнтів та запобігання втраті потенційних покупців через незручності використання сайту;
- інтеграція з платіжними системами: магазин повинен підтримувати різні платежі;
- управління запасами: інтернет-магазин повинен мати систему ефективного управління запасами товарів. Це включає моніторинг наявності товарів, автоматичне оновлення запасів, сповіщення про низький рівень запасів і можливість виконання попереднього замовлення;
- маркетинг та просування: варто враховувати можливості для маркетингу та просування продуктів у межах інтернет-магазину. Це можуть бути акції, знижки, програми лояльності, огляди та рейтинги товарів, а також інструменти для просування через соціальні мережі та інші онлайн-канали;
- аналітика та звітність: Важливо мати можливість відстежувати та аналізувати результати продажів, поведінку клієнтів та ефективність маркетингових кампаній. Інструменти аналітики допоможуть зрозуміти, як покращити ефективність магазину та приймати обґрунтовані рішення щодо розвитку бізнесу;
- підтримка клієнтів: інтернет-магазин повинен мати механізми для зв'язку з клієнтами, включаючи онлайн-чат, електронну пошту або телефонну підтримку. Це допоможе вирішувати питання клієнтів, надавати консультації та забезпечувати якісний сервіс.

Розглянемо також основні етапи розробки інтернет-магазину:

- а) аналіз вимог та планування:
 - визначення цілей та обсягу проекту;

- вивчення цільової аудиторії та ринкових умов;
 - визначення функціональних вимог до магазину;
 - розробка плану проекту та графіка робіт;
- б) проектування:
- розробка структури та архітектури магазину;
 - вибір технологічного стеку для реалізації магазину;
 - розробка дизайну та інтерфейсу користувача;
 - створення прототипів та макетів сторінок;
- в) розробка:
- реалізація функціональності магазину, включаючи реєстрацію користувачів, пошук товарів, кошик покупок, оформлення замовлення, оплату та доставку;
 - інтеграція зовнішніх сервісів, таких як платіжні шлюзи, системи управління запасами або доставки;
 - розробка адміністративної панелі для управління товарами, замовленнями, клієнтами та іншими аспектами магазину;
- г) тестування та відлагодження:
- перевірка функціональності та взаємодії елементів магазину;
 - виявлення та виправлення помилок та дефектів;
 - виконання тестових замовлень та перевірка роботи процесу оплати та доставки.

1.2 Аналіз аналогів інтернет-магазину з продажу автомобілів

Аналіз інтернет-магазинів з продажу автомобілів в Україні показує, що цей сегмент електронної комерції досить розвинений та конкурентний. Декілька відомих та популярних інтернет-магазинів автомобілів в Україні включають такі платформи:

а) Auto.ria.com (рисунок 1.1): цей інтернет-магазин є одним з найбільших та найпопулярніших в Україні. Він пропонує широкий вибір нових та б/у автомобілів різних марок та моделей. Auto.ria.com також надає користувачам можливість продати свій автомобіль;

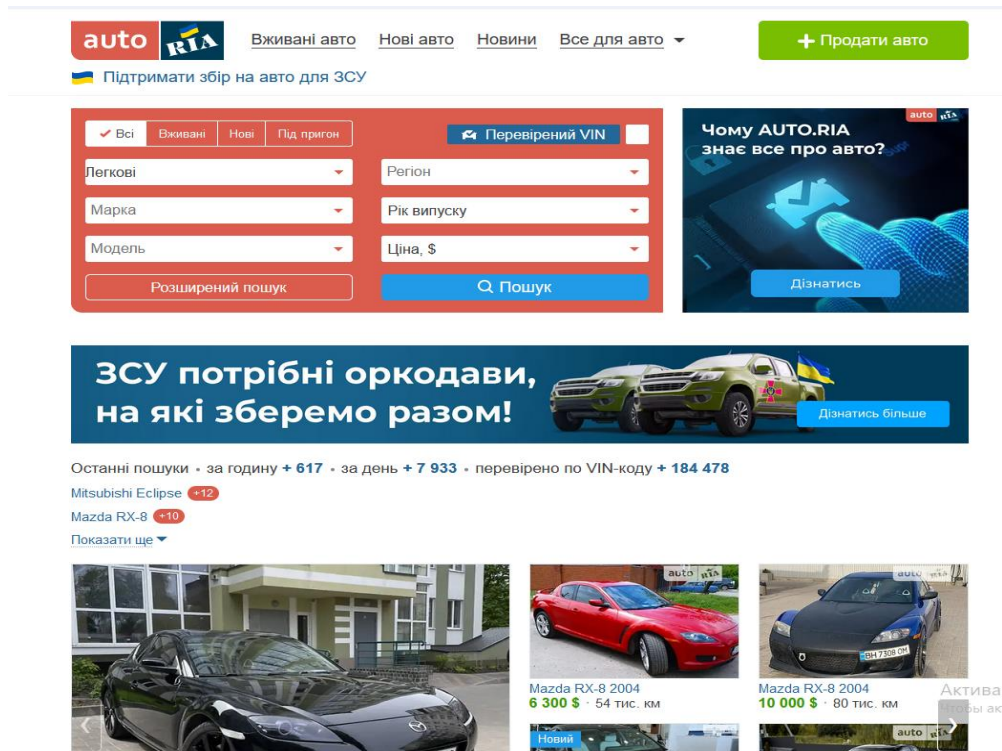


Рисунок 1.1 – Інтерфейс Auto.ria

б) Autoscout24.ua (рисунок 1.2): ця платформа спеціалізується на продажі б/у автомобілів та надає широкий вибір автомобілів від різних дилерів та приватних продавців. Autoscout24.ua також надає корисні функції пошуку та фільтрації, щоб допомогти користувачам знайти потрібний автомобіль;

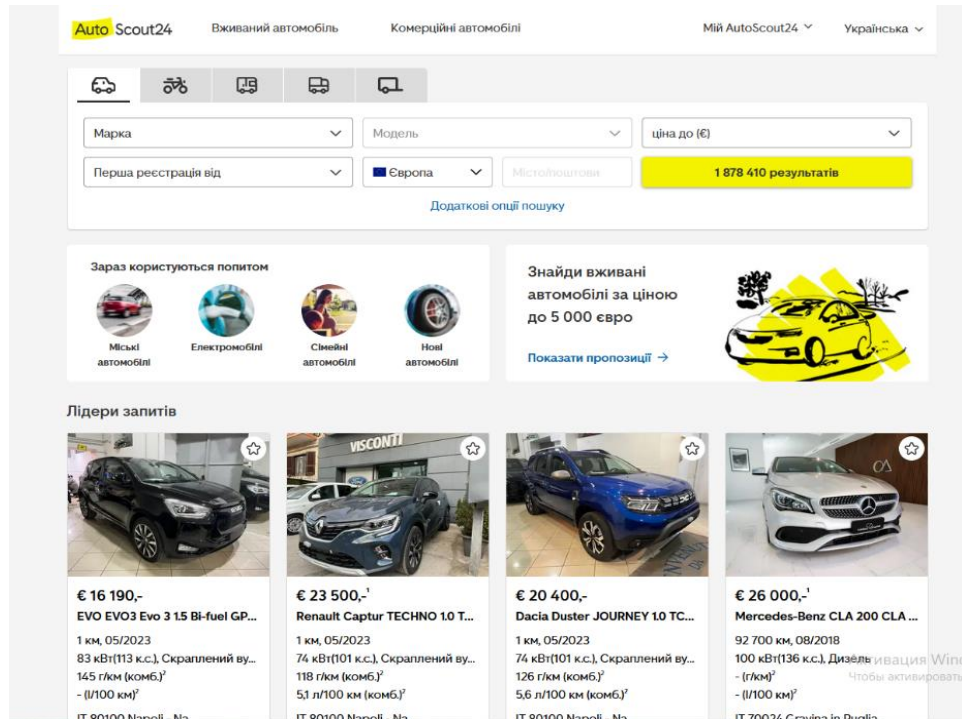


Рисунок 1.2 – Інтерфейс Autoscout24.ua

в) Slando.ua (рисунок 1.3): цей інтернет-магазин також належить до OLX Group та дозволяє користувачам розміщувати та шукати оголошення про продаж автомобілів. Slando.ua має велику кількість оголошень про різні типи автомобілів та варіанти цін;

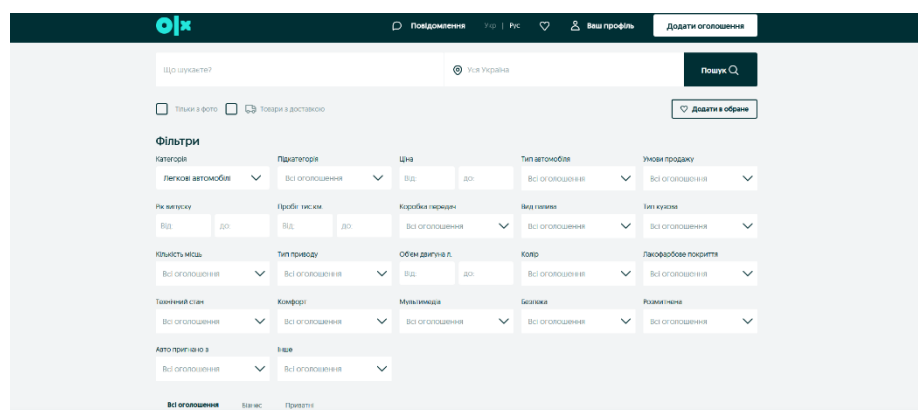


Рисунок 1.3 – Інтерфейс Slando.ua

г) RST.ua (рисунок 1.4): цей інтернет-магазин спеціалізується на продажі автомобілів бізнес-класу та ексклюзивних автомобілів. RST.ua пропонує широкий

вибір висококласних автомобілів відомих виробників, таких як BMW, Mercedes-Benz, Audi та інші. Вони пропонують розширені функціональні можливості пошуку за маркою, моделлю, роком випуску та іншими параметрами, що дозволяють користувачам швидко знайти бажаний автомобіль;

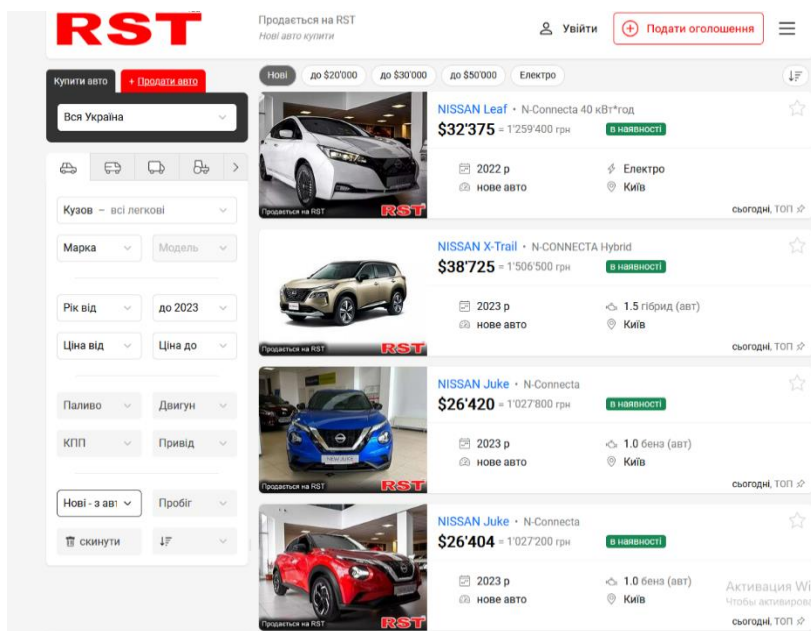


Рисунок 1.4 – Інтерфейс RST.ua

д) AutoSale.ua (рисунок 1.5): цей інтернет-магазин спеціалізується на продажі нових автомобілів в Україні. Вони працюють з автосалонами та дилерами різних марок, що дозволяє представляти широкий спектр автомобілів для покупців;

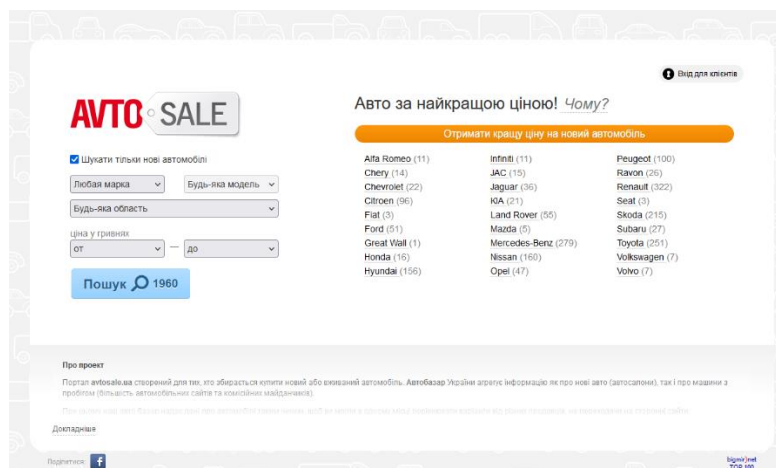


Рисунок 1.5 – Інтерфейс AutoSale.ua

е) UkrAutoprom.com: цей магазин спеціалізується на продажі нових та б/у автомобілів вітчизняного виробництва. UkrAutoprom.com пропонує широкий вибір автомобілів від відомих українських виробників, таких як ZAZ, Bogdan, LAZ та інші.

Дані магазини надають різноманітні можливості для покупців, включаючи пошук за різними параметрами, перевірку технічного стану автомобіля, можливість зручні та зв'язку з продавцями.

1.3 Специфікація вимог до програмного забезпечення інтернет-магазину з продажу автомобілів

Специфікація вимог до програмного забезпечення інтернет-магазину з продажу автомобілів може включати наступні елементи:

а) реєстрація користувачів:

- можливість реєстрації нових користувачів;
- авторизація за допомогою електронної пошти або соціальних мереж;
- зберігання особистої інформації користувача, такої як контактні дані і адреса доставки;

б) пошук і фільтрація автомобілів:

- можливість швидкого та зручного пошуку автомобілів за маркою, моделлю, роком випуску, ціновим діапазоном, типом трансмісії та іншими характеристиками;
- фільтрація результатів пошуку за допомогою різних параметрів;

в) деталі продукту:

- повна інформація про кожен автомобіль, включаючи фотографії, технічні характеристики, опис, історію обслуговування та стан автомобіля;
- можливість додати власні коментарі та оцінки до продукту;

г) кошик та оформлення замовлення:

- додавання товарів до кошика та зберігання їх до оформлення замовлення;

- можливість обрати різні варіанти оплати та доставки;
- перегляд та редагування замовлення перед підтвердженням;

д) інтеграція зовнішніх сервісів:

- інтеграція з платіжними шлюзами для здійснення безпечних онлайн-транзакцій;
- зв'язок з системами управління запасами, щоб відображати актуальну наявність товарів;
- інтеграція з системами доставки для розрахунку вартості та вибору способу доставки;

е) адміністративна панель:

- можливість керувати інтернет-магазином та каталогом автомобілів;
- додавання, редагування та видалення автомобілів у каталозі;
- керування замовленнями, перегляд статусу та опрацювання замовлень користувачів;
- управління користувачами, включаючи реєстрацію, блокування та редагування інформації про користувачів;
- генерація звітів та статистики про продажі, запаси та інші показники діяльності;

ж) мобільна версія та адаптивний дизайн:

- розробка мобільної версії інтернет-магазину або використання адаптивного дизайну, щоб забезпечити зручний доступ та перегляд каталогу автомобілів на різних пристроях;

з) безпека та захист даних:

- забезпечення безпеки особистих даних користувачів та захисту фінансових транзакцій;
- використання шифрування та інших заходів безпеки для запобігання несанкціонованому доступу до даних;

и) зворотній зв'язок та оцінки:

– можливість залишати відгуки, оцінки та коментарі до автомобілів та обслуговування;

– система збору та відображення відгуків користувачів, щоб допомогти іншим покупцям прийняти рішення.

Це вимоги, які можуть бути враховані при розробці програмного забезпечення інтернет-магазину з продажу автомобілів. Конкретні вимоги можуть відрізнятися в залежності від бізнес-потреб, розміру компанії та цільової аудиторії.

Висновки до розділу 1

В ході виконання першого розділу було проведено аналіз предметної області, визначено актуальність обраної теми . Розглянуто та проаналізовані сучасні аналоги

2 МОДЕЛІ, МЕТОДИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1 HTML5 – мова розмітки гіпертексту

HTML5 (HyperText Markup Language 5) (рисунок 2.1) є останньою версією стандарту HTML, який використовується для створення та відображення веб-сторінок. Він є мовою розмітки, що використовується для структурування та візуалізації вмісту на веб-сторінках.



Рисунок 2.1 – логотип HTML5

Кожна веб-сторінка повинна мати обов'язкову структуру, яка допомагає організувати та представити контент коректно. Основна структура HTML-документа включає наступні елементи:

- елемент `<html>`: визначає кореневий елемент HTML-документа;
- елемент `<head>`: містить метайнформацію про сторінку, таку як заголовок, мета теги та посилання на зовнішні файли CSS або JavaScript;
- елемент `<body>`: вміщує основний вміст сторінки, такий як текст,

зображення, посилання та інші елементи;

– блокові елементи: наведені раніше блокові елементи, такі як `<div>`, `<p>`, `<h1>`-`<h6>`, ``, ``, використовуються для організації та розмітки різних блоків контенту на сторінці, вони займають повний рядок та автоматично створюють новий блок нижче;

– рядкові елементи: рядкові елементи, такі як `<a>`, ``, `
`, `<code>`, `<input>`, використовуються для форматування тексту та вставки рядкових елементів всередині блокових елементів;

– непарні елементи: непарні елементи, як ``, не мають закриваючого тегу і використовуються для вставки зображень або інших мультимедійних елементів.

Використання відповідних тегів та їх вкладення допомагає створювати структурований і семантичний код HTML-сторінок. За допомогою CSS можна впливати на вигляд та оформлення цих елементів, надаючи їм стилі та властивості, які задовольняють дизайн сторінки та вимоги користувачів. Детальнішу інформацію про кожен елемент та його характеристики можна знайти в офіційній документації HTML.

Кожна сторінка повинна містити таку базову структуру (рисунок. 2.2).

```
index.html
src > index.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10
11 </body>
12 </html>
```

Рисунок 2.2 – Базова структура HTML

Тег `<!DOCTYPE html>` визначає тип документа і повідомляє браузеру, що це HTML5-документ. Тег `<html>` є кореневим елементом, який охоплює всі інші елементи сторінки. Елемент `<head>` використовується для конфігурації документа, такої як встановлення заголовка сторінки, вказівки щодо кодування, масштабування та інші метадані. Елемент `<body>` містить структуру основного контенту документа, такого як тексти, зображення, відео, аудіо та інші елементи. Існує безліч редакторів коду, які надають інструменти та методи для створення веб-сторінок. Найпопулярніші з них включають WebStorm, Visual Studio Code, Sublime Text та багато інших. Кожен редактор має свої унікальні особливості, і розробники можуть обирати той, який найкраще підходить для їх потреб та вподобань. Блокові елементи формують видимі блоки на сторінці. Вони починаються з нового рядка та займають повний доступний простір по горизонталі. Блокові елементи не можуть бути вкладеними всередину рядкових елементів, але іноді можуть бути вкладеними в інші елементи блокового рівня. Вони дозволяють організувати та групувати контент для досягнення бажаного макету та структури на сторінці.

HTML продовжує бути однією з найбільш актуальних мов розмітки для веб-розробки з рядом вагомих причин:

незамінність у веб-сфері: HTML є незамінною мовою для створення веб-сторінок, вона забезпечує основну структуру контенту, що дозволяє пошуковим системам та браузерам ефективно зчитувати та відображати сторінки;

- широка сумісність: HTML є стандартом у галузі веб-розробки і має високий рівень сумісності з різними браузерами, операційними системами та пристроями;
- простота та доступність: HTML має простий синтаксис, що робить його легким у вивченні та використанні;
- розширюваність: HTML постійно розвивається та оновлюється, що дозволяє впроваджувати нові функціональні можливості та стандарти;
- інтеграція з іншими технологіями: HTML легко поєднується з CSS для стилізації та зовнішнього вигляду сторінок, а також з JavaScript.

Загалом, HTML залишається ключовим інструментом у веб-розробці завдяки своїй незамінності, сумісності, простоті використання та можливостям розширення, що дозволяють створювати вражаючі та потужні веб-сайти та додатки.

Незважаючи на численні переваги, HTML також має деякі недоліки:

- відсутність динамічності: HTML зосереджений на статичній розмітці та відображенні контенту;
- обмежені можливості макетування: HTML сам по собі не забезпечує широкі можливості стилізації та макетування веб-сторінок;
- сумісність з різними браузерами: незважаючи на стандартизацію, різні браузери можуть трактувати HTML-код по-різному;
- обмежена можливість розробки додатків: HTML, як мова розмітки, має свої обмеження в порівнянні з повноцінними мовами програмування.

Хоча HTML має свої недоліки, вони зазвичай компенсуються використанням інших технологій та здатністю HTML поєднуватися з ними для створення більш потужних та інтерактивних веб-додатків.

2.2 Таблиця стилів CSS3

CSS3 (Cascading Style Sheets 3) є останньою версією мови CSS, яка використовується для стилізації та оформлення веб-сторінок. (рисунок. 2.3).



Рисунок 2.3 – логотип CSS3

Без використання стилів (CSS), будь-який HTML-документ, незалежно від кількості елементів, буде суттєво обмеженим у візуальному представленні. CSS, або каскадні таблиці стилів, визначають зовнішній вигляд документа. Стил у CSS представляє собою правило, яке вказує веб-браузеру, як формувати елемент. Форматування може включати встановлення кольору фону елемента, встановлення кольору та типу шрифту і так далі. Визначення стилю складається з двох частин: селектора, який вказує на елемент, та блоку оголошення стилю - набору команд, які встановлюють правила форматування (рисунок 2.4).

```
# main.css
styles > # main.css > ...
1  div{
2    background-color: green;
3    width: 100px;
4    height: 60px;
5  }
6
7
```

Рисунок 2.4 – Оголошення стилю

У даному випадку селектором є `div`. Цей селектор вказує, що цей стиль буде застосовуватися до всіх елементів `div`. Після селектора в фігурних дужках йде блок оголошення стилю. Між відкриваючою і закриваючою фігурними дужками визначаються команди, які вказують, як форматувати елемент. Кожна команда складається з властивості та значення. Наприклад, у наступному виразі (рисунок 2.5): `background-color` - це властивість, а `green` - значення. Властивість визначає конкретний стиль. У CSS існує багато різних властивостей. Наприклад, `background-color` визначає колір фону. Після двокрапки йде значення для цієї властивості. У вищезазначеній команді властивості `background-color` встановлюється значення `green`. Іншими словами, для фону елемента встановлюється зелений колір.

```
background-color: green;
```

Рисунок 2.5 – Приклад властивості та значення

Іноколи для одних і тих самих елементів потрібна різна стилізація. У такому випадку ми можемо використовувати класи. Для визначення селектора класу в CSS перед назвою класу ставиться крапка (рисунок 2.6).

```
7  
8   .green{  
9     background-color: greenyellow;  
10  }  
11
```

Рисунок 2.6 – Селектор класу

Назва класу може бути будь-якою. Наприклад, у даному випадку назва класу - "green". Проте, при цьому використання в назві класу дозволяється букви, цифри, дефіси та підкреслення, причому назва класу має починатися обов'язково з літери. Також слід враховувати регістр імен: назви "green" і "GREEN" будуть представляти різні класи. Після визначення класу ми можемо його застосувати до елемента за допомогою атрибута `class` (рисунок 2.7).

```
<div class="green"></div>>
```

Рисунок 2.7 – Застосування класу

Для ідентифікації унікальних елементів на веб-сторінці використовуються ідентифікатори, які визначаються за допомогою атрибутів id. Наприклад, на сторінці може бути головний блок або заголовок (рисунок 2.8).

```
<div id="group"></div>
```

Рисунок 2.8 – Застосування атрибуту

Визначення стилів для ідентифікаторів відбувається аналогічно визначенню стилів для класів, за винятком того, що замість крапки використовується символ хештегу # (рисунок 2.9).

```
#group{  
  height: 180px;  
  background-color: # cornflowerblue;  
}
```

Рисунок 2.9 – Ідентифікатор

2.3 Мова програмування JavaScript

JavaScript є невід'ємною складовою сучасних веб-сторінок і вважається однією з найпопулярніших та найважливіших мов програмування для веб-розробки. Її використання дозволяє створювати динамічні та інтерактивні веб-сторінки, розширювати функціональність веб-додатків та забезпечувати взаємодію з користувачами(рисунок 2.10).



Рисунок 2.10– Логотип JavaScript

JavaScript є високорівневою мовою програмування, яка підтримує різні підходи, такі як імперативний, функціональний та подієво-орієнтований. Вона характеризується динамічною типізацією і використовується для написання послідовних операцій або скриптів. Оскільки JavaScript зазвичай інтерпретується, а не компілюється, виконання коду не вимагає додаткових програм або процесу компіляції в інший рівень кодування.

Початково мова була відома як LiveScript, але з метою привернення більшої уваги до популярної мови програмування Java, вона була перейменована на JavaScript. Обрана назва мала залучити увагу та зацікавленість, оскільки мова Java була широко відомою. Після впровадження JavaScript, багато компаній висловили намір використовувати цю мову в своїх продуктах. Початково JavaScript була створена як скриптова мова для веб-розробки та взаємодії з користувачем на рівні браузера. З плином часу JavaScript стала потужною та багатофункціональною мовою програмування. Вона отримала стандартизацію через ECMAScript, яка визначила стандарти для мови. Сучасні версії JavaScript, такі як ECMAScript 6 (ES6) та ECMAScript 2020, мають численні нововведення та покращення, які полегшують розробку веб-додатків.

JavaScript має великий функціонал, який робить її потужною та універсальною мовою програмування. Основні функції включають:

- робота з анімацією та графікою: JavaScript дозволяє створювати анімаційні ефекти, слайдери, графіки та інші візуальні елементи на веб-сторінках. З використанням бібліотек, таких як jQuery або HTML5 Canvas, можна створювати складні візуальні ефекти та інтерактивні графічні додатки;

- розробка веб-додатків: JavaScript використовується для створення повноцінних веб-додатків, включаючи фронтенд та бекенд частини. З використанням фреймворків, таких як React, Angular або Node.js, можна створювати масштабовані та ефективні веб-додатки з розширеною функціональністю;

- маніпуляцію елементами веб-сторінки: JavaScript дозволяє змінювати вміст, стилі, атрибути та розміщення елементів на веб-сторінці. Це дозволяє створювати динамічні ефекти та змінювати вигляд сторінки залежно від дій користувача;

- реагування на події: JavaScript дозволяє додавати обробники подій, такі як клік, наведення, введення даних тощо, це дозволяє створювати інтерактивні елементи та взаємодіяти з користувачем;

- взаємодія з сервером: JavaScript може взаємодіяти з сервером за допомогою AJAX (асинхронний JavaScript та XML) для обміну даними без перезавантаження сторінки. Це дозволяє створювати динамічні веб-додатки та отримувати або надсилати дані на сервер без затримок;

- робота зі збереженими даними: JavaScript може зберігати дані на боці клієнта за допомогою локального сховища, такого як localStorage або cookies. Це дозволяє зберігати стан, налаштування або інші важливі дані без необхідності перезавантаження сторінки;

- валідація форм: JavaScript дозволяє перевіряти та валідувати дані, введені користувачем в форми, перед їх відправленням на сервер, це допомагає забезпечити правильність та повноту даних, що надсилаються;

– робота з анімацією та графікою: JavaScript дозволяє створювати анімаційні ефекти, слайдери, графіки та інші візуальні елементи на веб-сторінках. З використанням бібліотек, таких як jQuery або HTML5 Canvas, можна створювати складні візуальні ефекти та інтерактивні графічні додатки.

Серед усіх можливостей JavaScript можна виділити і декілька недоліків:

– швидкість виконання: у деяких випадках JavaScript може працювати повільніше порівняно з іншими мовами програмування, особливо при обробці великої кількості даних або виконанні складних обчислень;

– проблеми з масштабовністю: при розробці великих проєктів JavaScript може стикатися з проблемами масштабовності та підтримки коду. Відсутність строгої типізації та інших механізмів контролю може зробити код вразливим до помилок та ускладнити розуміння та підтримку кодової бази;

– чутливість до помилок: JavaScript допускає помилки, які можуть призвести до непередбачуваної поведінки і некоректних результатів. Неправильне використання може спричинити збої виконання скрипту;

– різна підтримка браузером: різні браузери можуть мати різну підтримку стандарту JavaScript та його функцій, це може створювати проблеми з переносимістю коду між браузерами та вимагати додаткових зусиль для забезпечення сумісності;

– обмеження безпеки: JavaScript виконується в контексті веб-сторінки та має обмеження безпеки. Він не має прямого доступу до операційної системи та обмежений у взаємодії з іншими веб-сторінками, це зроблено для запобігання можливим зловживанням та збереження безпеки користувача.

Незважаючи на ці недоліки, JavaScript залишається важливою та широко використовуваною мовою програмування для веб-розробки через свою потужну функціональність та можливості взаємодії з користувачем.

2.4 Бази даних

2.4.1 MySQL

MySQL є реляційною базою даних, яка використовує мову запитів SQL (Structured Query Language). Вона є однією з найпопулярніших систем керування базами даних (СКБД) і використовується для зберігання, керування та отримання даних.



Рисунок 2.11 – Логотип MySQL

MySQL був розроблений з фокусом на швидкість, надійність і простоту використання. Він пропонує широкий набір функцій, які дозволяють ефективно працювати з базами даних. Деякі основні можливості MySQL включають:

- створення та управління базами даних: MySQL дозволяє створювати нові бази даних та керувати існуючими. Ви можете визначати таблиці, поля, індекси та інші складові бази даних;
- мова запитів SQL: MySQL використовує стандартну мову запитів SQL для взаємодії з базою даних. Ви можете виконувати запити для створення, читання, оновлення та видалення даних, а також для виконання складних операцій, таких як з'єднання таблиць, сортування і групування даних;
- транзакції та конкурентний доступ: MySQL підтримує транзакції, що дозволяє виконувати набір операцій як єдину атомарну операцію. Він також забезпечує конкурентний доступ до даних, дозволяючи багатьом користувачам одночасно працювати з базою даних без блокування;
- безпека та аутентифікація: MySQL має механізми для захисту бази даних, включаючи систему ролей і прав доступу. Ви можете встановлювати обмеження доступу до даних та виконувати аутентифікацію користувачів;
- широкі можливості розширення: MySQL дозволяє розширювати функціональність бази даних за допомогою різних додаткових модулів і

розширень. Це дає можливість використовувати спеціалізовані функції, забезпечувати реплікацію даних, забезпечувати безпеку тощо.

2.4.2 PhpMyAdmin

При розробці веб-проектів або міграції сайтів доступ до бази даних є необхідним. PhpMyAdmin є популярним інструментом для роботи з базами даних MySQL або MariaDB. Цей веб-додаток з відкритим кодом написаний на мові PHP і має графічний веб-інтерфейс, який дозволяє керувати базами даних через браузер.

За допомогою PhpMyAdmin ви можете керувати сервером MySQL, виконувати SQL-запити та переглядати та редагувати вміст таблиць баз даних. Цей інструмент дуже зручний для веб-розробників, оскільки дозволяє керувати базами даних з будь-якого комп'ютера, підключеного до Інтернету, без потреби вводити SQL-команди, завдяки інтуїтивному інтерфейсу. Він також не вимагає встановлення додаткового програмного забезпечення, оскільки працює через веб-браузер.

PhpMyAdmin широко використовується в сучасній веб-розробці, оскільки розробники постійно оновлюють його, враховуючи останні нововведення у світі баз даних MySQL. Багато українських хостинг-провайдерів використовують PhpMyAdmin як панель управління, щоб надати своїм клієнтам можливість керувати їх базами даних.

Окрім PhpMyAdmin, існують інші інструменти для роботи з базами даних, такі як MySQL Workbench, Adminer та інші. Вибір конкретного інструменту залежить від особистих уподобань та потреб проекту.

2.5 PHP

PHP (PHP: Hypertext Preprocessor - "PHP: препроцесор гіпертексту") - це скриптова мова програмування, яка використовується для розробки застосунків та веб-сайтів. Наразі вона підтримується майже всіма хостинг-компаніями та є одним

з лідерів серед мов програмування для написання динамічних веб-сторінок. Наші хостинг-сервери також підтримують цю мову [17].

PHP - це мова програмування з відкритим вихідним кодом, розроблена для вебу, яка може бути вбудована в HTML-код. PHP є легким для вивчення, тому вже через кілька годин ознайомлення з його функціоналом можна почати писати прості скрипти мовою PHP.

Історія PHP починається з 1995 року, коли Расмус Лердорф (англ. Rasmus Lerdorf) створив простий застосунок мовою Perl, який аналізував відвідування користувачами його резюме на веб-сайті. Коли декілька людей вже користувалися цим застосунком, а число зацікавлених зростало, Лердорф назвав своє творіння "Personal Home Page Tools" (версія 1) та опублікував його для безкоштовного завантаження. Саме з цього моменту почалася неймовірна популярність PHP [18].

Основне призначення PHP - це обробка даних на веб-сервері. Після виконання на сервері PHP генерує HTML, який відправляється до клієнтського браузера, що дозволяє створювати динамічний контент. PHP може взаємодіяти з базами даних, обробляти форми, керувати файлами на сервері та виконувати багато інших завдань на рівні сервера.

Основні особливості PHP включають:

- синтаксис: PHP базується на синтаксисі мови C, що полегшує його вивчення та розуміння для програмістів, які знайомі з мовами C, C++ або Java;
- вбудована підтримка баз даних: PHP має вбудовану підтримку різних типів баз даних, включаючи MySQL, PostgreSQL, SQLite та інші. Це дозволяє легко взаємодіяти з базами даних та виконувати операції, такі як зчитування, запис і оновлення даних;
- розширені можливості: PHP має велику кількість вбудованих функцій та розширень, які спрощують роботу з веб-серверами, файлами, текстом, зображеннями та іншими завданнями. Це дозволяє розробникам швидко вирішувати різноманітні задачі та скорочує час розробки;

- переносимість: PHP підтримується багатьма операційними системами, такими як Windows, Linux, macOS та іншими. Код PHP може працювати на будь-якому веб-сервері, який підтримує PHP;

- велике співтовариство: PHP має широке співтовариство розробників, що означає, що завжди є багато документації, підручників, форумів та розширень, доступних для допомоги та підтримки.

Недоліки PHP включають:

- продуктивність: у порівнянні з деякими іншими мовами програмування PHP може бути менш ефективним у плані продуктивності. Він виконується на рівні сервера, що може призводити до затримок при роботі з великими навантаженнями;

- безпека: неправильна конфігурація або неуважність можуть призвести до потенційних проблем безпеки. У PHP є історія певних вразливостей, тому необхідно дотримуватися найкращих практик безпеки при розробці PHP-застосунків.

Загалом, PHP є популярною мовою програмування для розробки веб-сайтів та додатків. Вона має широкі можливості та активну спільноту розробників, що допомагає вирішувати завдання різної складності.

Висновки до розділу 2

В ході виконання другого розділу БКР було розглянуто та проведено аналіз мов програмування які будуть використані в ході програмної реалізації інтернет-магазину

3 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОГО ВЕБЗАСТОСУНКУ

3.1 Макет користувацького інтерфейсу

При розробці макету користувацького інтерфейсу необхідно враховувати потреби та вимоги користувачів, щоб забезпечити зручну та легку навігацію. Макет повинен відображати логіку функцій та можливостей веб-додатку, а також відповідати ергономічним принципам та дизайн-стандартам.

При розробці макету інтерфейсу для інтернет-магазину з продажу автомобілів можуть бути включені такі основні елементи:

- шапка: верхня частина сторінки, де розташовані логотип магазину, основне меню навігації, пошукова рядок та інші важливі елементи;
- банер або слайдер: великий графічний елемент, який привертає увагу користувачів і відображає промоційні пропозиції, нові автомобілі або особливі акції;
- категорії або фільтри: можливість вибору категорій автомобілів або використання фільтрів для точного підбору автомобіля за параметрами, такими як марка, модель, ціновий діапазон і т.д.;
- список продуктів: відображення автомобілів у вигляді списку або сітки, зображення автомобілів, коротка інформація про них та ціна;
- сторінка товару: детальна сторінка автомобіля з більш повною інформацією, включаючи фотографії, технічні характеристики, опис, ціну, варіанти обладнання та інші додаткові деталі;
- кошик: функціональний блок, що дозволяє користувачам додавати обрані автомобілі до кошика для подальшого оформлення замовлення;
- форма замовлення: сторінка, на якій користувач може заповнити свої контактні дані, вибрати метод оплати та доставки, а також підтвердити замовлення;

– футер: нижня частина сторінки, де розташована додаткова інформація, така як контактні дані магазину, посилання на соціальні мережі, посилання на додаткові сторінки.

Ці елементи допоможуть створити зручний та інтуїтивно зрозумілий макет інтерфейсу для інтернет-магазину з продажу автомобілів. Вони сприятимуть зручній навігації та допоможуть користувачам легко знайти та придбати бажані автомобілі.

На рисунку 3.1 можемо бачити макет першої сторінки інтернет магазину.

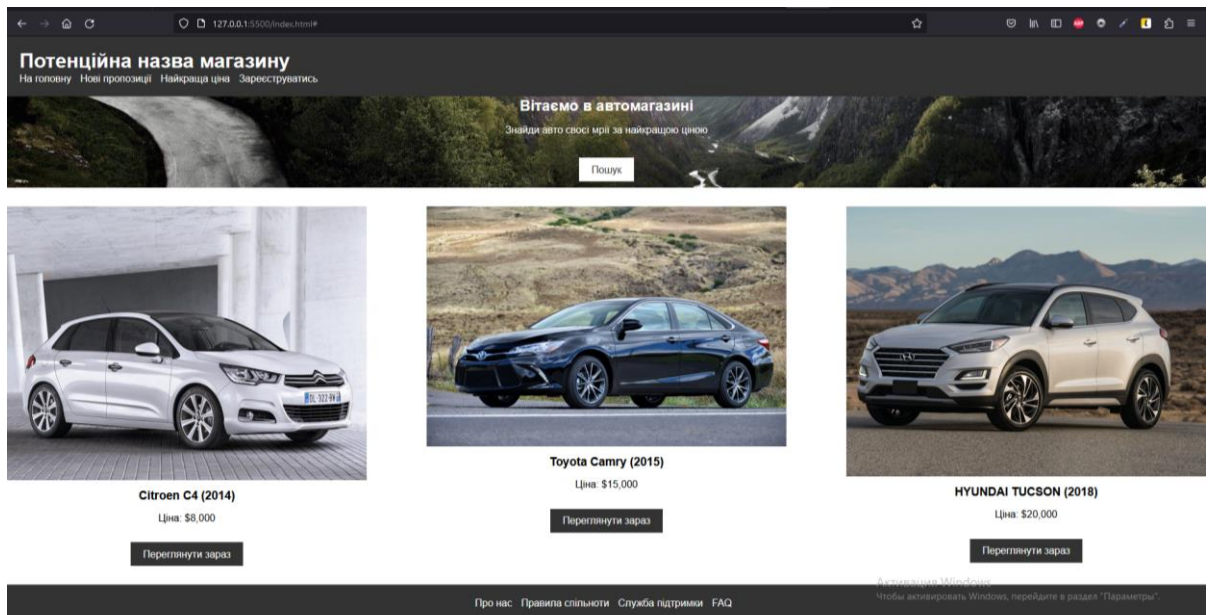


Рисунок 3.1– Макет головної сторінки магазину

Реєстрація та авторизація в інтернет-магазині мають велику важливість і принесуть багато переваг як власникам магазину, так і користувачам. Ось кілька ключових причин, чому ці функції є необхідними:

– особисті облікові записи: реєстрація дозволяє користувачам створити особистий обліковий запис в інтернет-магазині. Це дає можливість зберігати та управляти персональною інформацією. Користувачі можуть зручно входити в свій обліковий запис, щоб здійснювати покупки та використовувати інші функції магазину;

- безпека даних: авторизація забезпечує захист особистої інформації користувача, так як доступ до облікового запису здійснюється за допомогою унікальних облікових даних, наприклад, логіна та пароля. Це допомагає уникнути несанкціонованого доступу до особистих даних та покупок користувача;
- особисті налаштування: після авторизації користувач може налаштувати свої особисті вподобання;
- зручність покупок: реєстрація та авторизація роблять процес покупок в інтернет-магазині швидким та зручним.

В цілому, реєстрація та авторизація дозволяють покращити взаємодію магазину з користувачами, забезпечити безпеку та персоналізацію, спростити процес покупок та створити позитивний досвід для клієнтів.

Потенційна назва

введіть ваше ім'я

введіть ваш Email

придумайте пароль

повторіть пароль

Зареєструватись

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Рисунок 3.2– Макет сторінки реєстрації

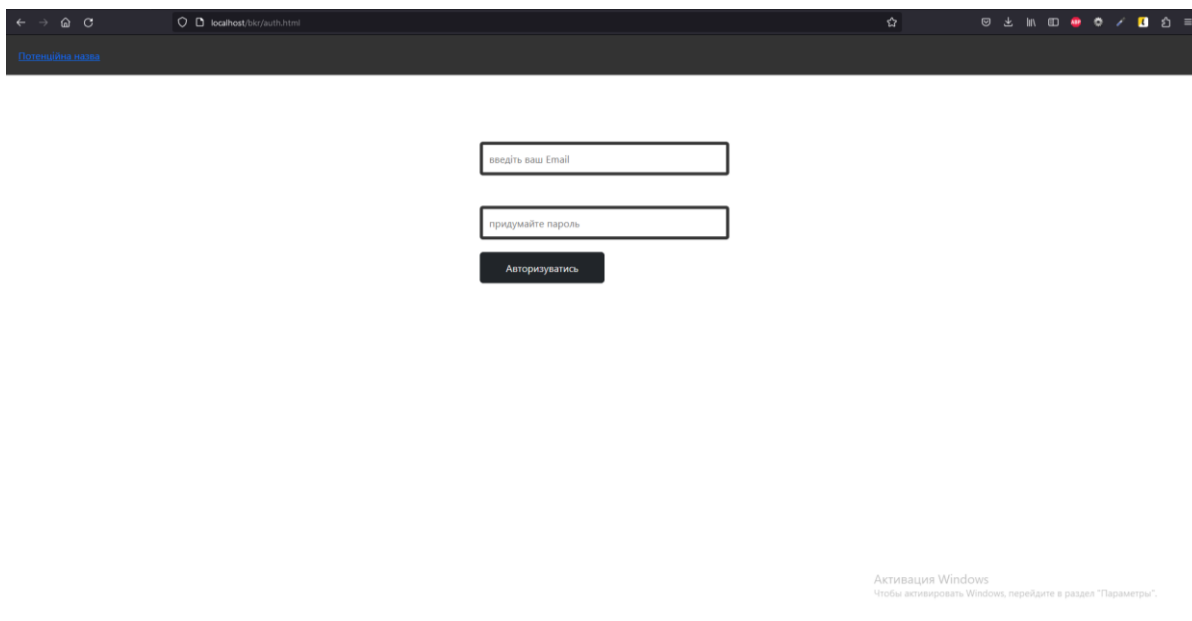


Рисунок 3.3– Макет сторінки авторизації

Оголошення, створені користувачами на сайті інтернет-магазину з продажу автомобілів, мають велику важливість і декілька переваг для самого магазину, а також для потенційних покупців.

Ось декілька причин, чому оголошення користувачів є важливими:

- розширення асортименту: оголошення, створені користувачами, дозволяють розширити асортимент автомобілів, які доступні в магазині. Користувачі можуть пропонувати різноманітні марки, моделі та варіанти автомобілів, що збільшує вибір для покупців. Це дозволяє магазину стати більш привабливим для різних клієнтів з різними потребами та бюджетами;

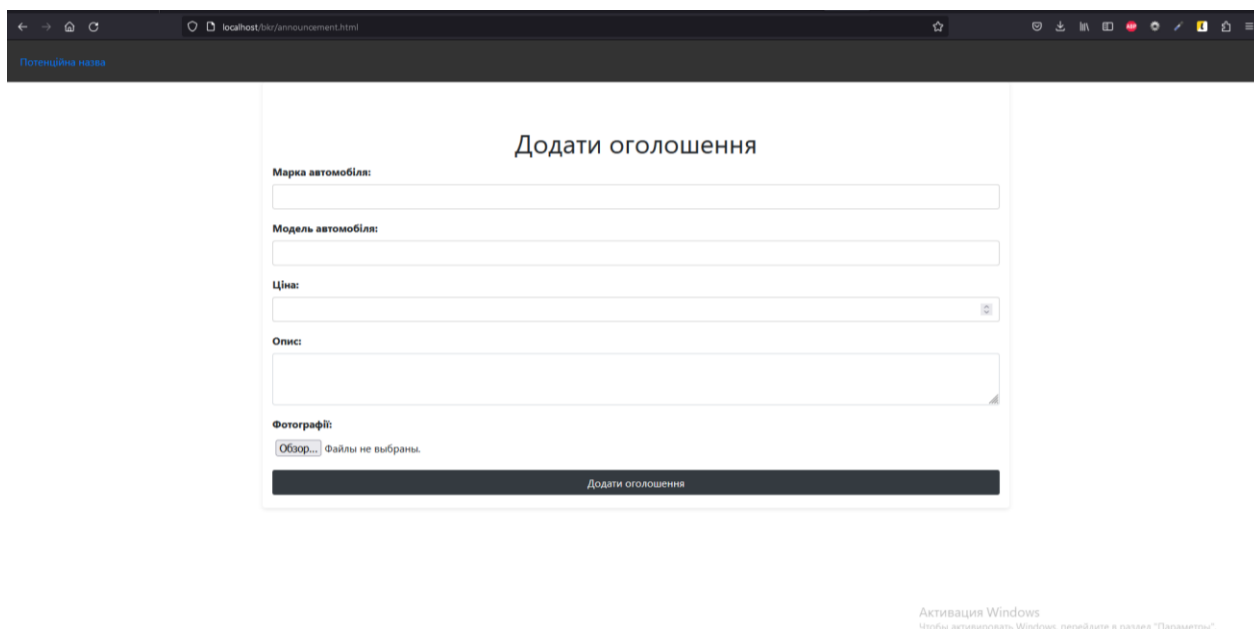
- розширення ринку: користувачі можуть розміщувати оголошення про продаж автомобілів з різних регіонів, що дозволяє магазину привертати клієнтів з усієї країни або навіть з-за кордону. Це розширює потенційну аудиторію і збільшує шанси на успішну продаж автомобілів;

- достовірність інформації: користувачі, які розміщують оголошення, зазвичай надають детальну інформацію про автомобіль, включаючи його технічні характеристики, стан, історію обслуговування тощо. Це допомагає покупцям зробити обґрунтований вибір і надає їм довіру до інформації, яка надається магазином;

– залучення більш активної спільноти: оголошення користувачів створюють можливість для взаємодії та обговорення між продавцями і потенційними покупцями. Користувачі можуть задавати питання, висловлювати свої думки та ділитися своїм досвідом. Це сприяє формуванню активної спільноти навколо магазину, покращує залучення клієнтів і може привести до більшій кількості продажів;

– зниження витрат: використання оголошень користувачів дозволяє магазину зменшити витрати на закупівлю автомобілів для подальшої реалізації. Замість придбання автомобілів напряму від виробників або дилерів, магазин може співпрацювати з користувачами, які продають свої власні автомобілі. Це може призвести до зниження витрат на придбання автомобілів та підвищення рентабельності бізнесу.

Усі ці фактори роблять користувацькі оголошення важливою складовою інтернет-магазину з продажу автомобілів. Вони створюють додаткові можливості для розширення бізнесу, залучення нових клієнтів та поліпшення загального досвіду покупки автомобілів.



Потенційна назва

localhost/biz/announcement.html

Додати оголошення

Марка автомобіля:

Модель автомобіля:

Ціна:

Опис:

Фотографії:
 Файлы не выбраны.

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Рисунок 3.4 – Макет сторінки додання оголошень

Сторінка асортименту в інтернет-магазині з продажу автомобілів є однією з найважливіших складових, яка впливає на ефективність та привабливість магазину для потенційних покупців. Ось декілька причин, чому сторінка асортименту є такою важливою:

- перші враження та навігація: сторінка асортименту є першим контактом клієнта з продуктами, які пропонує магазин. Важливо, щоб ця сторінка була привабливою та легко навігуюною. Чітко представлені фотографії, короткі описи та зручні фільтри допоможуть клієнтам швидко зорієнтуватися в пропозиціях і знайти потрібний автомобіль;

- широкий вибір та різноманітність: сторінка асортименту має демонструвати широкий вибір автомобілів з різних марок, моделей, років випуску, характеристик та цінних категорій. Різноманітність в асортименті дозволяє задовольнити потреби різних клієнтів з різними бюджетами та вимогами до автомобілів;

- детальна інформація про кожен автомобіль: кожен автомобіль на сторінці асортименту має бути супроводжений детальною інформацією, такою як технічні характеристики, історія обслуговування, стан автомобіля, інформація про попередніх власників тощо. Це допомагає покупцям зробити обґрунтований вибір, порівняти автомобілі та визначити, який з них найкраще відповідає їхнім потребам.

- зображення та відео: візуальні матеріали, такі як фотографії високої якості та відеоогляди автомобілів, є надзвичайно важливими для сторінки асортименту. Якісні фотографії, які демонструють зовнішній вигляд автомобіля, його інтер'єр та деталі, допомагають клієнтам отримати уявлення про автомобіль і викликати інтерес до нього;

- фільтри та пошук: сторінка асортименту має мати функціональність фільтрів та пошуку, яка дозволяє клієнтам швидко знаходити автомобілі, що відповідають їхнім вимогам. Фільтри можуть включати критерії, такі як марка, модель, рік випуску, цінний діапазон, пробіг тощо. Це спрощує процес пошуку та збільшує зручність для клієнтів;

– коментарі та відгуки: наявність коментарів та відгуків від попередніх клієнтів на сторінці асортименту допомагає підтвердити надійність і якість автомобілів, а також створює додаткову довіру до магазину. Позитивні відгуки можуть стимулювати інтерес клієнтів та спонукати їх до покупки.

Сторінка асортименту є важливим інструментом для привертання та утримання клієнтів в інтернет-магазині з продажу автомобілів. Якісне представлення продуктів, детальна інформація та зручність навігації сприяють успіху бізнесу та задоволенню потреб покупців.

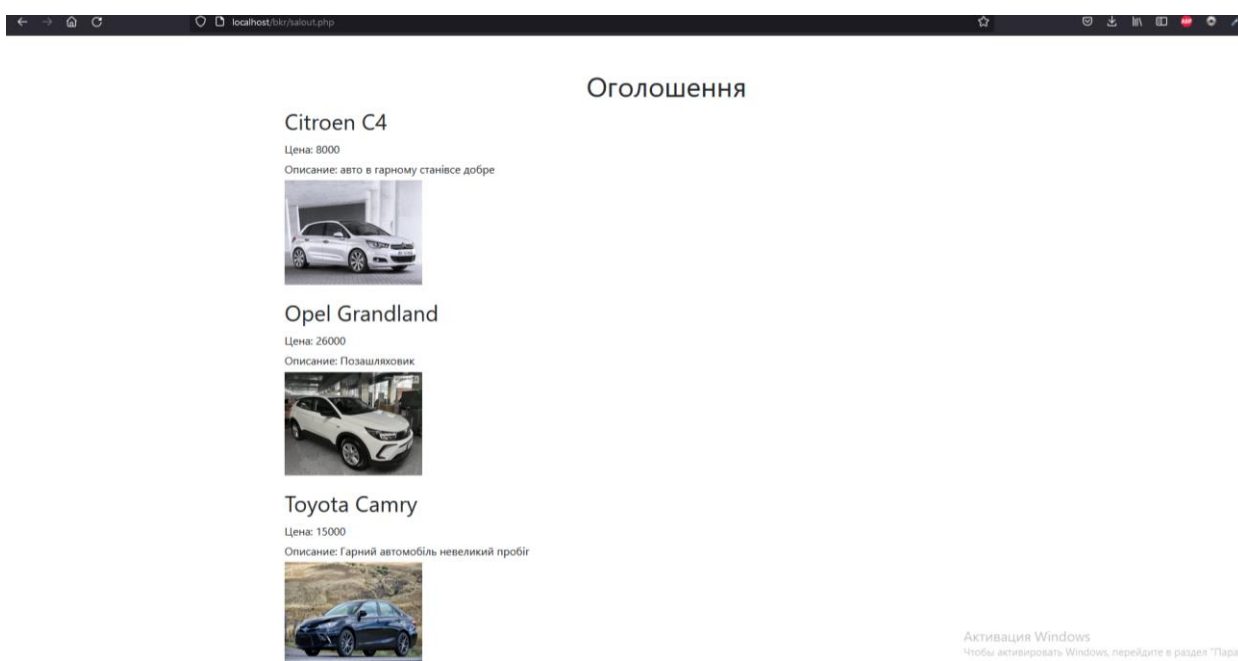


Рисунок 3.5– Макет сторінки асортименту оголошень

Особистий кабінет в інтернет-магазині з продажу автомобілів є важливою функціональною складовою, яка надає клієнтам зручність, персоналізацію та контроль над їхнім досвідом покупок. Ось декілька причин, чому особистий кабінет є важливим:

– особисті налаштування та збереження інформації: клієнти можуть налаштувати свій особистий кабінет, додавши свої вподобання, зберігаючи дані для швидкого оформлення замовлень, встановлюючи сповіщення про нові автомобілі або знижки. Це дозволяє зекономити час при наступних покупках та забезпечує більш персоналізований досвід;

– керування замовленнями: в особистому кабінеті клієнти можуть відстежувати свої замовлення, переглядати статус доставки, отримувати повідомлення про оновлення та спілкуватися зі службою підтримки. Це надає зручність та контроль над процесом покупок.

Особистий кабінет у інтернет-магазині з продажу автомобілів забезпечує зручність, персоналізацію та контроль над процесом покупок для клієнтів. Це стимулює повторні покупки, підвищує задоволеність клієнтів та допомагає розвивати довгострокові взаємовідносини з клієнтами.

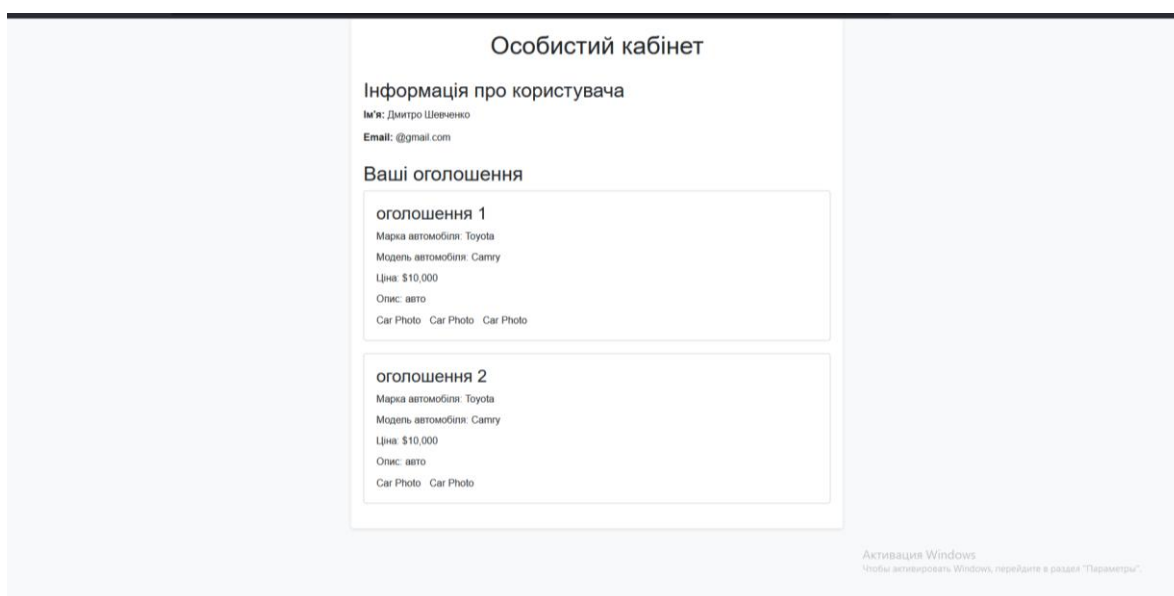


Рисунок 3.6 – Макет сторінки особистого кабінету

3.2 Візуальна карта веб-застосунку

Візуальна карта сайту (sitemap) є графічним зображенням структури веб-сайту. Вона відображає ієрархію сторінок та їх зв'язки між собою. Візуальна карта може мати різні формати, включаючи дерево, схему або таблицю, залежно від вимог проекту.

Основне призначення візуальної карти сайту полягає в полегшенні навігації для користувачів та пошукових систем. Вона надає зручний спосіб швидкого огляду всіх доступних сторінок на сайті і допомагає виявити логічні зв'язки між ними. Користувачам стає легше зорієнтуватися на сайті, знайти потрібну

інформацію та перейти до відповідних сторінок. Пошукові системи, такі як Google, використовують візуальну карту для сканування сторінок і визначення, які сторінки пов'язані між собою. Це сприяє кращому індексуванню сайту пошуковими системами і поліпшує його видимість у пошукових результатах.

Основні переваги використання візуальної карти сайту:

- допомагає користувачам зорієнтуватися на сайті та знайти потрібну інформацію. Користувачі можуть швидко оцінити обсяг і структуру сайту та зрозуміти, які сторінки доступні;
- покращує індексацію сайту пошуковими системами. Візуальна карта допомагає пошуковим системам зрозуміти структуру та зв'язки між сторінками, що поліпшує процес індексації;
- служить важливим інструментом для розробників сайтів при плануванні структури сайту. Візуальна карта допомагає визначити логічну організацію контенту та зв'язків між сторінками. Вона може бути використана як основа для розробки ієрархії сторінок та навігаційних елементів;
- допомагає виявити проблеми з навігацією та недоступністю сторінок. Візуальна карта дозволяє швидко переглянути всі сторінки і виявити "порожні" сторінки, які не мають зв'язків з іншими частинами сайту або є важкодоступними для користувачів.

Візуальна карта сайту є корисним інструментом для покращення навігації, пошукової оптимізації та планування структури веб-сайту. Вона допомагає як користувачам, так і розробникам зрозуміти і покращити структуру сайту для зручної навігації та покращення впливу на пошукові системи.

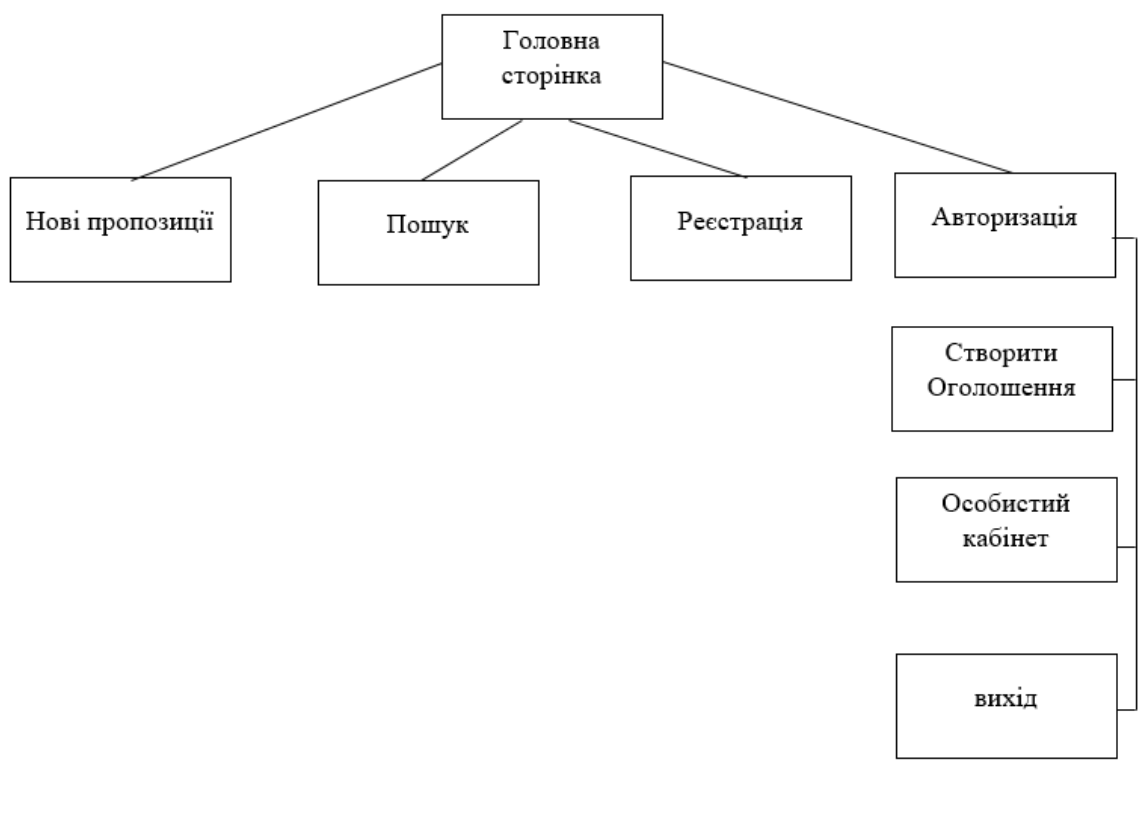


Рисунок 3.7 Карта веб застосунку

3.3 Сценарії використання

Сценарії використання є детальним описом послідовності подій між користувачами та системою. Вони визначають, як користувачі взаємодіють з системою, які дії вони виконують та які результати очікують. Ці сценарії допомагають виявити можливості системи, її цінність та показати, як вона може розв'язувати реальні проблеми.

Основні цілі сценаріїв використання включають:

- розуміння функціональних потреб користувачів: Сценарії допомагають розробникам зрозуміти, як користувачі сприймають і використовують систему; Вони виявляють дії, які користувачі виконують та їхню взаємодію з системою.
- визначення функціональних вимог: Сценарії допомагають встановити потреби користувачів і визначити функціональні вимоги до системи. Вони

вказують, які функції та можливості повинна мати система, щоб задовольнити потреби користувачів;

– оцінка взаємодії: Сценарії дозволяють оцінити, наскільки ефективно система задовольняє потреби користувачів. Вони допомагають виявити можливі проблеми взаємодії та знайти шляхи для поліпшення.

Отже, сценарії використання є важливим інструментом для розробки та оцінки системи, дозволяючи зорієнтуватися на потреби користувачів і створити ефективну взаємодію між користувачами та системою.

Сценарії використання описують, як користувачі будуть взаємодіяти з вебсайтом та як система реагуватиме на їхні дії. Вони включають послідовність кроків, необхідних для досягнення цілей користувача. Сценарії використання мають велике значення, оскільки допомагають уточнити поведінку системи та виявити можливі проблеми. Вони також визначають цілі, що допомагають оцінити цінність та складність системи.

У системі присутні актори - зовнішні сутності, що взаємодіють з системою через прецеденти. Сама система виконує дії, визначені в кожному сценарії. Діаграма використання програмного забезпечення є засобом візуалізації цих акторів та їхньої взаємодії з системою.

Сценарії використання є важливим інструментом для розуміння та специфікації взаємодії користувачів із системою. Вони дозволяють визначити послідовність дій, виконуваних користувачами, та відповіді системи на ці дії. Це сприяє виробленню чіткого уявлення про функціональність системи та допомагає виявити можливі проблеми з розробкою та взаємодією.

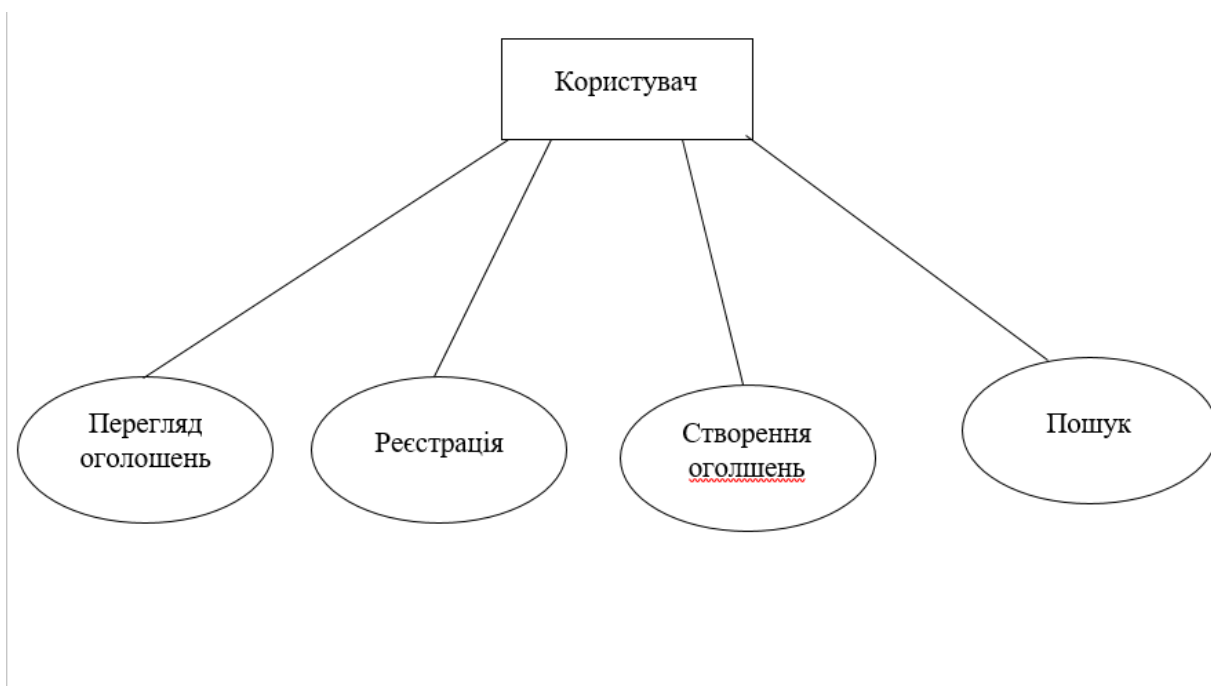


Рисунок 3.8 – Діаграма використання

Висновки до розділу 3

В ході виконання третього розділу було проведено аналіз та виділено основний функціонал вебзастосунку. Аналіз спрямований на розуміння потреб користувачів і визначення ключових можливостей, які вебзастосунок повинен надавати для задоволення цих потреб.

Були розроблені макети, у яких відображаються розташування елементів на сторінках вебзастосунку. Створено карту веб застосунку

4 ТЕСТУВАННЯ РОЗРОБЛЕНОГО ВЕБЗАСТОСУНКУ

4.1 Опис інтерфейсу вебзастосунок

По завершенню використання та розробки макетів, вебзастосунок можна використовувати. При потраплянні на вебзастосунок нас зустрічає головна сторінка (рис. 4.1). Застосунок зроблений зрозуміло сприйнятним для користувача. Були додані усі необхідні функції для зручного користування.

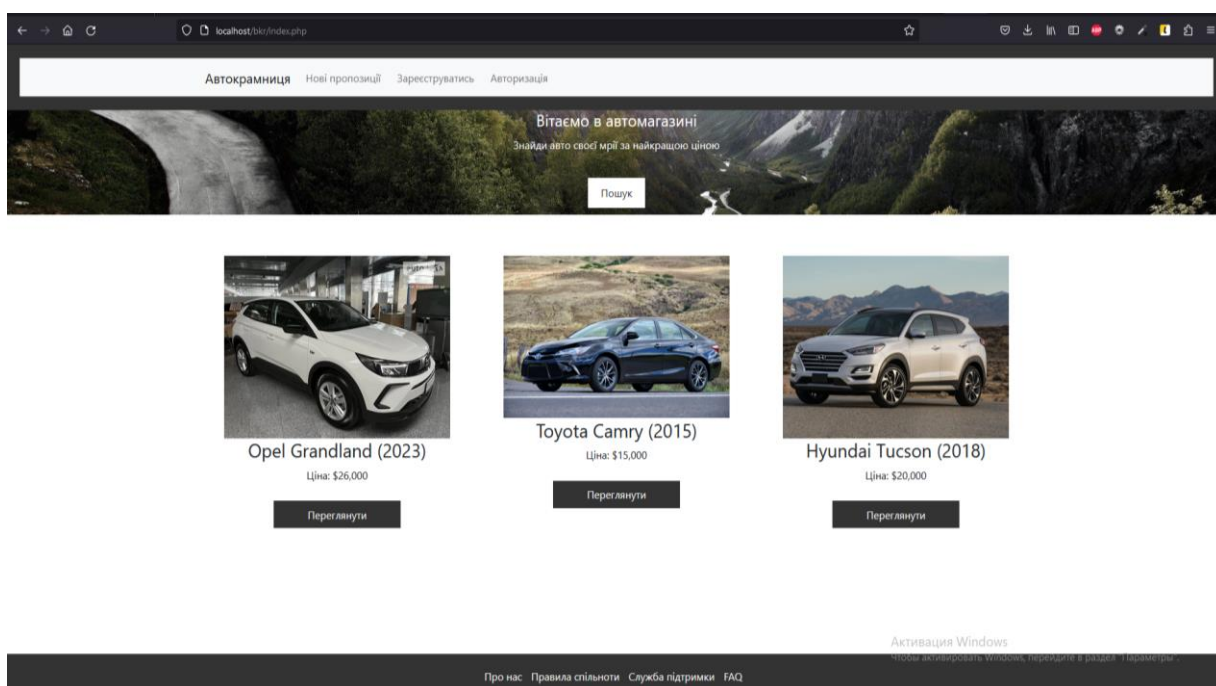


Рисунок 4.1 – Головна сторінка вебзастосунок

Якщо зайти на сторінку «Пропозиції» можна побачити оголошення залишені користувачами.

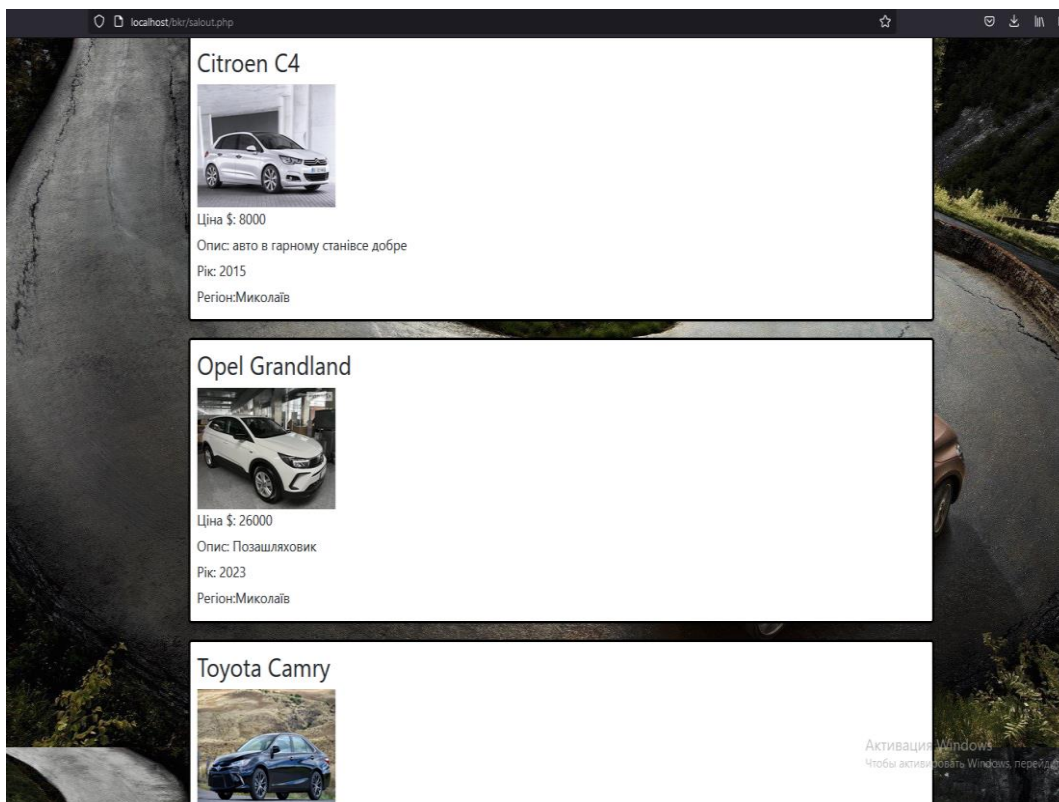


Рисунок 4.2 – сторінка оголошень

Коли користувач переглянув оголошення які його цікавлять він повернеться на головну сторінку і побачить що в вебзастосунку ще можна зареєструватись. Після натискання на кнопку «Зареєструватись» (рис. 4.3) перед користувачем з'явиться форма реєстрації (рис. 4.4).

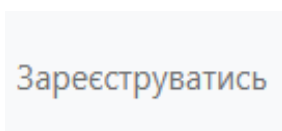


Рисунок 4.3 – кнопка реєстрації

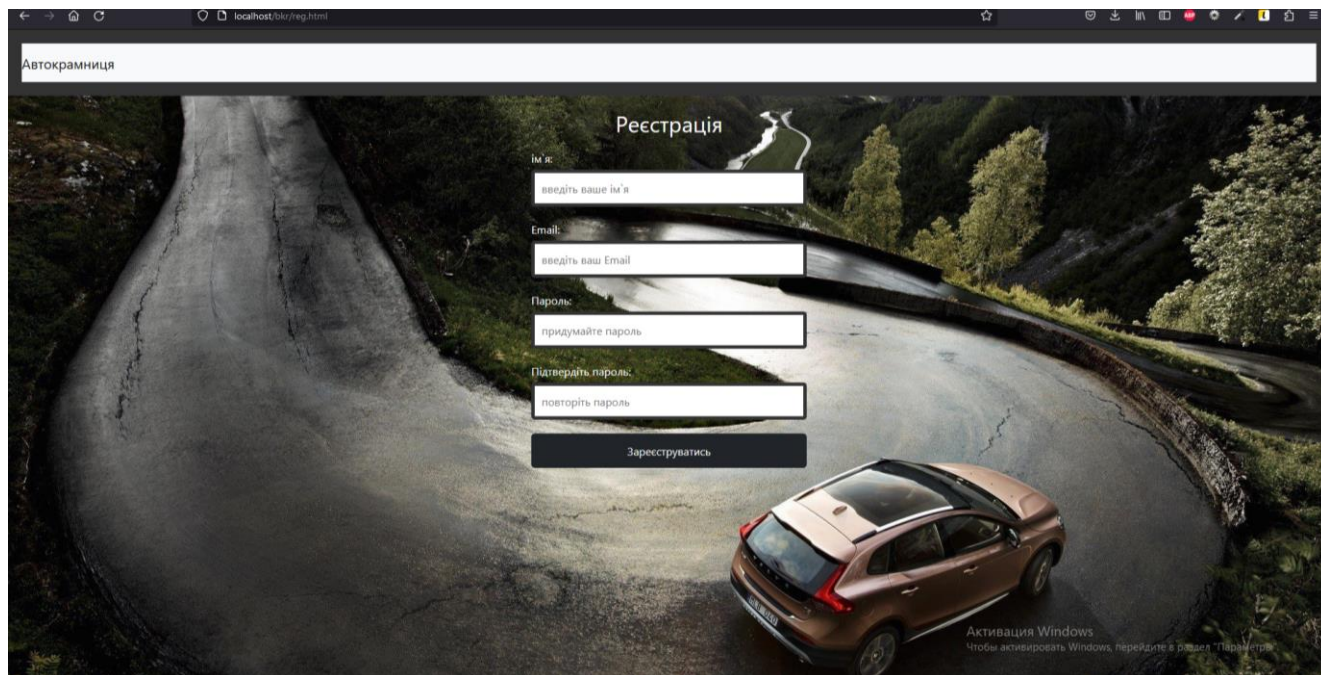


Рисунок 4.4 – сторінка реєстрації

Після того як користувач зареєструвався його перенаправить на сторінку авторизації (рис. 4.5) де він зможе спокійно авторизуватись в свій обліковий запис.

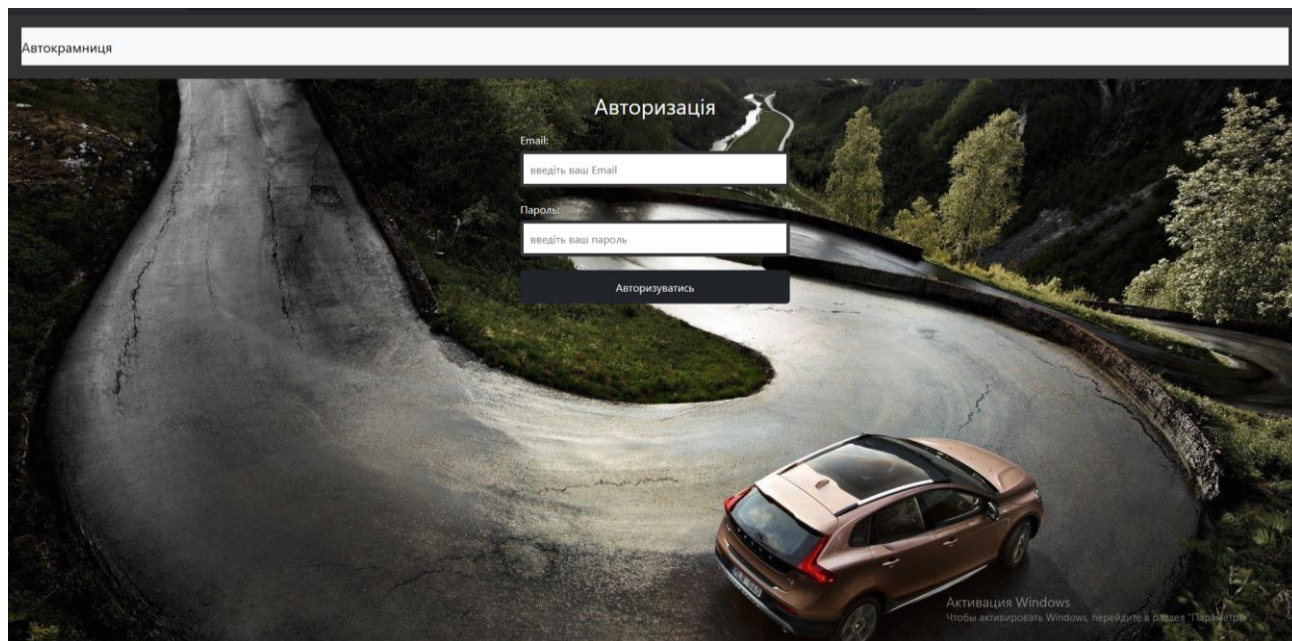


Рисунок 4.5 – сторінка авторизації

Як тільки користувач авторизувався він знову повертається на головну але тепер у нього з'являється більше можливостей він може не тільки дивитись оголошення а може ще й створювати їх. Після авторизації у користувача на верхній панелі з'являються нові пункти (рис. 4. 6)

Створити оголошення Особистий кабінет Вийти

Рисунок 4.6 – пункти

Натиснувши на пункт «Створити оголошення» користувач потрапить на сторінку де зможе створити оголошення про продаж авто по наявним критеріям. Його запит збережеться в базі даних і автоматично виведеться в наявні пропозиції.

Автокранниця

Додати оголошення

Марка автомобіля:

Модель автомобіля:

Ціна:

Опис:

Фотографії:

Обзор... Файлы не выбраны.

Рік:

Регіон:

Додати оголошення

Активация Windows

Рисунок 4.7 – форма додавання оголошень

Програмна реалізація даної форми та підключення до бази даних наведена на (рис. 4.8- 4.9).

```

<div class="container" >
  <h1 class="mt-5">Додати оголошення</h1>
  <form action="sale.php" method="POST" enctype="multipart/form-data">
    <div class="form-group">
      <label for="make">Марка автомобіля:</label>
      <input type="text" class="form-control" id="make" name="make" required>
    </div>

    <div class="form-group">
      <label for="model">Модель автомобіля:</label>
      <input type="text" class="form-control" id="model" name="model" required>
    </div>

    <div class="form-group">
      <label for="price">Ціна:</label>
      <input type="number" class="form-control" id="price" name="price" required>
    </div>

    <div class="form-group">
      <label for="description">Опис:</label>
      <textarea class="form-control" id="description" name="description" required></textarea>
    </div>

    <div class="form-group">
      <label for="photo">Фотопрограії:</label>
      <input type="file" class="form-control-file" id="photo" name="photo[]" multiple>
    </div>

    <button type="submit" class="btn btn-dark">Додати оголошення</button>
  </form>
</div>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

```

Рисунок 4.8 – реалізація форми «Створення оголошень»

```

sale.php
<?php
$servername = "localhost";
$username = "root";
$password = "root";
$dbname = "reg-db";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Помилка підключення: " . $conn->connect_error);
}

$userId = $_COOKIE['user_id'];

$make = $_POST['make'];
$model = $_POST['model'];
$price = $_POST['price'];
$description = $_POST['description'];

$targetDir = "uploads/";
$uploadedPhotos = [];

foreach ($_FILES['photo']['tmp_name'] as $index => $tmpName) {
    $targetFile = $targetDir . basename($_FILES['photo']['name'][$index]);
    move_uploaded_file($tmpName, $targetFile);
    $uploadedPhotos[] = $targetFile;
}

$sql = "INSERT INTO carsale (make, model, price, description, image, user_id)
VALUES ('$make', '$model', '$price', '$description', '" . implode(", ", $uploadedPhotos) . "', '$userId')";

if ($conn->query($sql) === TRUE) {
    echo "Объявление успешно добавлено!";
} else {
    echo "Помилка при добавлении объявления: " . $conn->error;
}

$conn->close();
?>

```

Рисунок 4.9 – форми «Створення оголошень»

Також вебзастосунок має сторінку користувача де відображається інформація про користувача така як ім'я , електронна адреса та оголошення які створив користувач, для зручності їх подальшого редагування (рис 4.10).

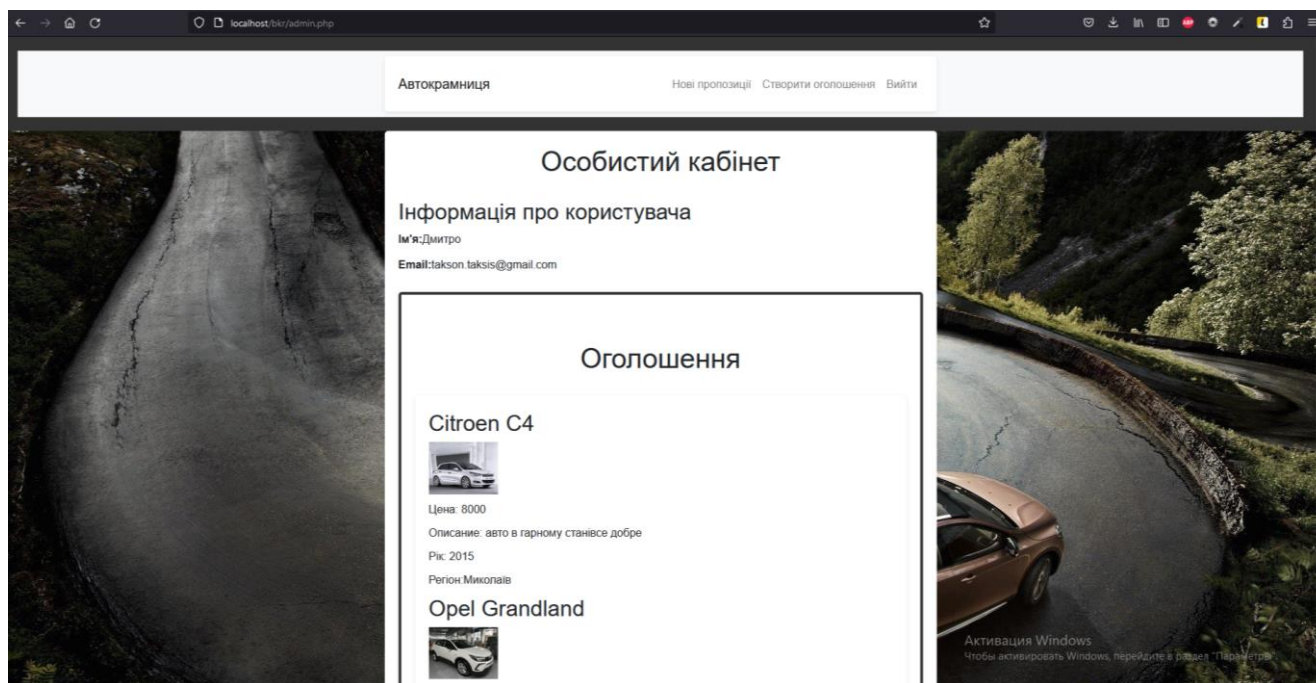


Рисунок 4.10 – Сторінка користувача

Форма внесення змін в оголошення створені конкретним користувачем через базу даних (рис 4.11).

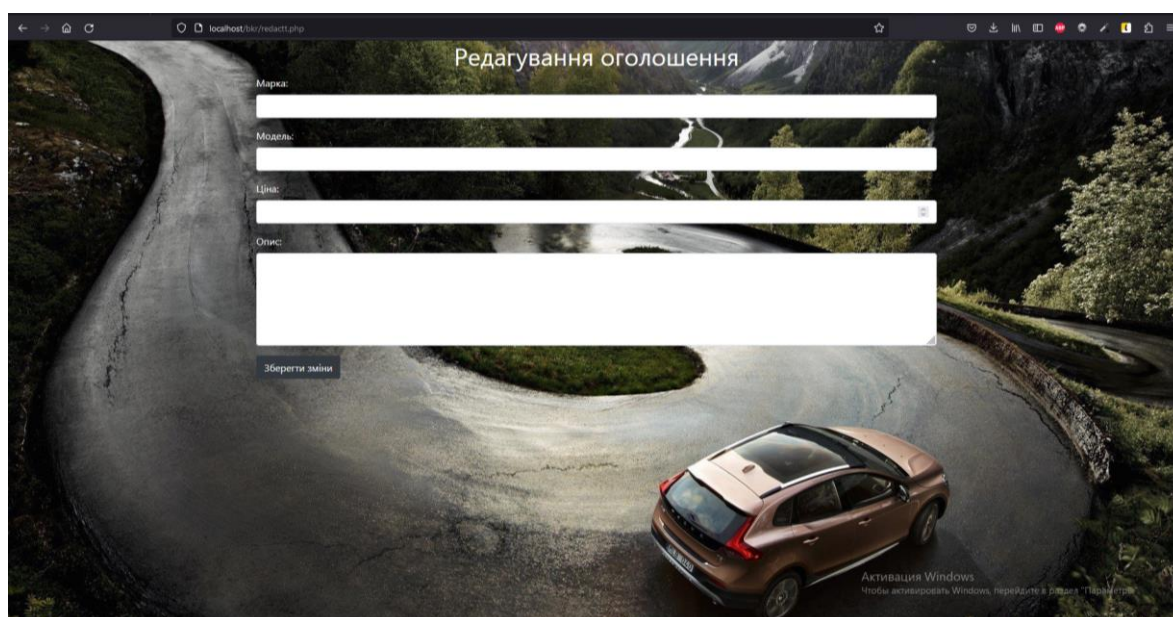


Рисунок 4.11 – Форма редагування оголошення

На сайті присутній пошук автомобілей по критеріям (рис. 4.12).

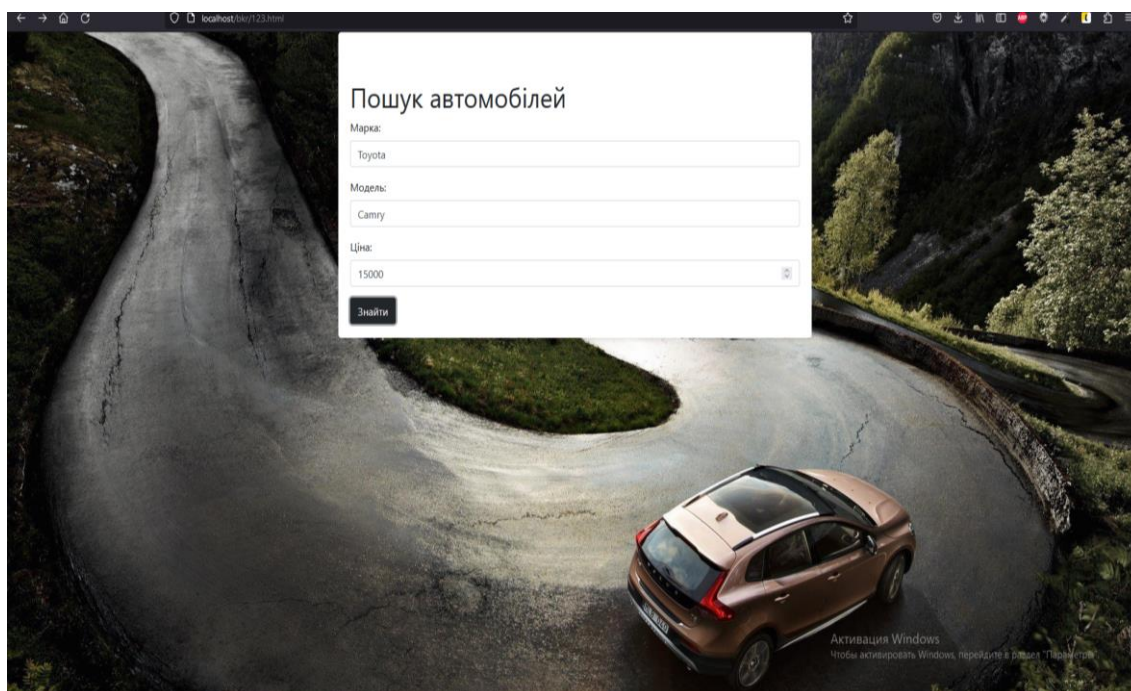


Рисунок 4.12 – Форма пошуку

По даним критеріям здійснюється пошук по існуючим на сайті оголошенням.

4.2 Фізична модель бази даних

Для створення фізичної моделі бази даних було використано MAMP Server та PhpMyAdmin з пакетом для MySQL.

MAMP дозволяє легко встановлювати, налаштовувати та керувати різними серверними компонентами, такими як Apache, PHP, MySQL, FTP-сервер і багато інших.

Основна мета MAMP Server полягає в спрощенні процесу налаштування та управління веб-сервером для розробки і тестування веб-додатків локально на вашому комп'ютері. Вона надає зручний інтерфейс, який дозволяє вам вмикати та вимикати різні серверні модулі, налаштовувати параметри, переглядати журнали подій і багато іншого.

За допомогою MAMP Server ви можете створювати локальні веб-сервери для розробки і тестування своїх веб-проектів перед їх публікацією на живому сервері.

Вона також підтримує роботу з різними версіями серверних програм та дозволяє швидко перемикається між ними.

Користування MAMP Server спрощує роботу з веб-сервером, особливо для початківців, які не мають глибоких знань у сфері серверного адміністрування. Вона надає зручні інструменти для управління сервером і дозволяє ефективно використовувати його для розробки і тестування веб-додатків.



Рисунок 4.13 – MAMP logo

PhpMyAdmin - це веб-програма, яка надає зручний графічний інтерфейс для керування базами даних MySQL. Розроблена на мові PHP, вона дозволяє розробникам і адміністраторам з легкістю виконувати різні операції з базами даних через веб-браузер.

Основна мета phpMyAdmin полягає в спрощенні управління базами даних MySQL. За допомогою цієї програми можна створювати нові бази даних і таблиці, виконувати SQL-запити, імпортувати та експортувати дані, налаштовувати користувачів та їх привілеї, а також здійснювати операції з резервними копіями та багато іншого.

PhpMyAdmin забезпечує зручний інтерфейс, що дозволяє зручно взаємодіяти з базами даних без необхідності використовувати складні інструменти або командний рядок. Цей інструмент особливо корисний для розробників веб-

додатків, які використовують базу даних MySQL, оскільки він спрощує процес розробки, відладки та керування базами даних. Завдяки простому та інтуїтивно зрозумілому інтерфейсу, phpMyAdmin надає доступ до різноманітних функцій MySQL, що полегшує взаємодію з даними.

На (рис. 4.14 4.15) зображені таблиці бази даних які були використані під час виконання роботи.

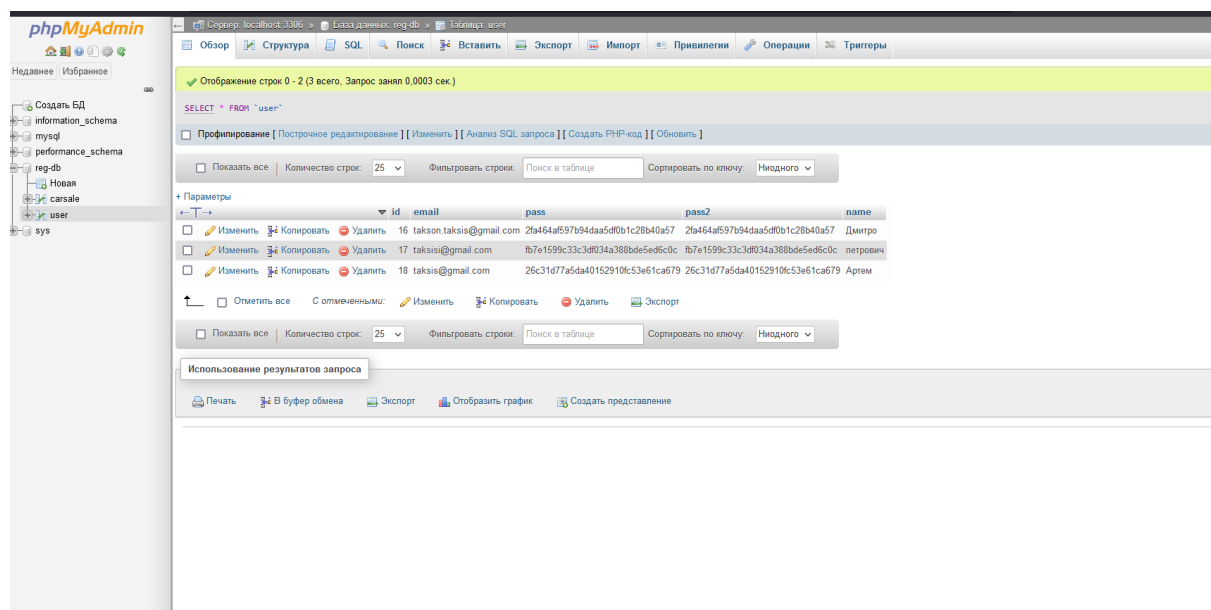


Рисунок 4.14 – Таблиця user

В даній таблиці зберігаються данні зареєстрованих людей наприклад:

- id користувача;
- ім'я;
- електронна пошта;
- пароль(захешований).

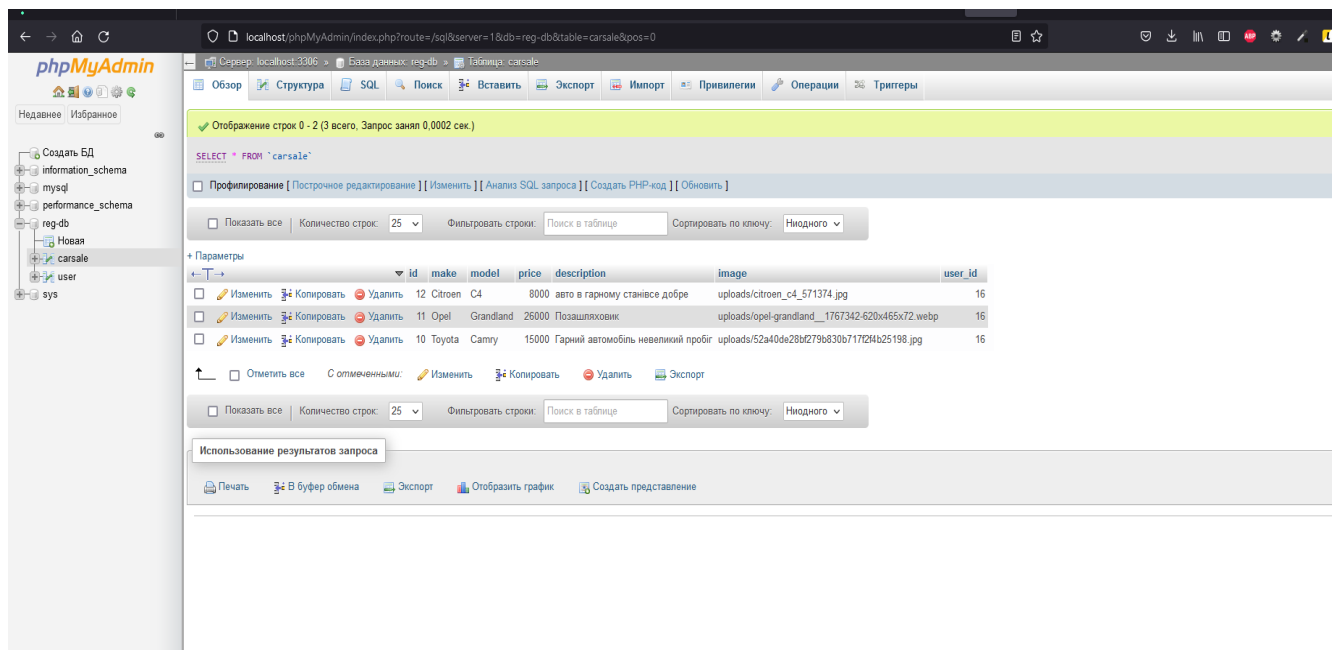


Рисунок 4.14 – Таблиця carsale

Таблиця містить в собі дані про створені оголошення містить поля:

- марка ;
- модель;
- ціна;
- опис;
- зображення;
- id оголошення;
- id користувача який створив дане оголошення.

4.3 Інструкція для користувача вебзастосунку

Нижче наведені загальні інструкції для користувачів веб-застосунку з продажу автомобілів.

а) реєстрація:

- відкрийте веб-застосунок та знайдіть кнопку або посилання для реєстрації;

- натисніть на кнопку або перейдіть за посиланням для реєстрації нового облікового запису;
- заповніть необхідні поля реєстраційної форми, такі як ім'я, прізвище, електронна пошта та пароль;
- після введення всіх необхідних даних натисніть кнопку "Зареєструватися" або подібну кнопку для завершення реєстрації;

б) авторизація:

- перейдіть на головну сторінку веб-застосунку;
- знайдіть кнопку або посилання для входу в систему;
- натисніть на кнопку або перейдіть за посиланням для входу до системи;
- введіть свої облікові дані, такі як електронна пошта і пароль;
- натисніть кнопку "Увійти" або подібну кнопку для авторизації у системі;

в) перегляд асортименту автомобілів:

- після успішного входу в систему, зазвичай ви потрапите на сторінку з асортиментом автомобілів;
- Прогляньте доступні автомобілі та перегляньте їх детальну інформацію, таку як опис, фотографії тощо;

г) розміщення оголошення про продаж автомобіля:

- знайдіть кнопку або посилання "Розмістити оголошення" або подібну опцію;
- натисніть на кнопку або перейдіть за посиланням для розміщення оголошення;
- заповніть необхідну інформацію про автомобіль, наприклад, марку, модель, рік випуску, стан, ціну, контактні дані тощо;
- додайте фотографії автомобіля, що допоможуть привернути увагу покупців;
- дати натисніть кнопку "Розмістити" або подібну кнопку для розміщення оголошення;

д) керування особистим кабінетом:

- більшість веб-застосунків з продажу автомобілів надають особистий кабінет для користувачів;
- використовуйте особистий кабінет для перегляду та керування своїми оголошеннями, редагування інформації про профіль, перегляду повідомлень від зацікавлених покупців тощо;

е) навігація по вебзастосунку:

- використовуйте головне меню або навігаційні посилання на сайті для переходу до різних розділів та функцій вебзастосунку;

ж) забезпечення безпеки:

- використовуйте міцний пароль для свого облікового запису та нерозголошуйте його іншим особам;
- будьте обережні під час використання вебзастосунку на публічних або невідтримуваних пристроях;
- регулярно перевіряйте оновлення програмного забезпечення

вебзастосунку та дотримуйтеся рекомендацій щодо безпеки, які надаються розробниками.

Це загальні кроки, які можуть бути присутніми у веб-застосунку з продажу автомобілів. Зверніться до документації чи інструкцій конкретного веб-застосунку для отримання більш детальної інформації та адаптації до конкретних функціональних можливостей системи.

Висновки до розділу 4

Було розроблено веб застосунок з продажу автомобілів, створений на спостереження та висновках.

Функціонал готового застосунку має розділи :

- головна;
- пропозиції;
- реєстрація;
- авторизація;
- створення оголошення

- особистий кабінет;
- редагування оголошень;
- пошук.

В ході роботи було показано, як кожен розділ працює та як користувач взаємодіє із ним.

ВИСНОВКИ

У даній бакалаврській роботі було проведено детальне дослідження вебзастосунків з продажу автомобілів і їх вплив на сучасний автомобільний ринок. Отримані результати дозволяють зробити наступні висновки.

Вебзастосунки з продажу автомобілів стали необхідним інструментом для автопродавців і покупців, оскільки вони забезпечують зручну платформу для огляду, порівняння та придбання автомобілів. Вони дозволяють покупцям отримати доступ до широкого асортименту автомобілів з усього світу і зробити вибір на основі детальної інформації, фотографій та відео.

Одним з головних переваг вебзастосунків з продажу автомобілів є можливість здійснювати покупку в будь-який зручний час і з будь-якого місця, що сприяє збільшенню зручності та доступності для покупців. Крім того, такі засоби дозволяють ефективно проводити порівняльний аналіз автомобілів, зокрема щодо ціни, характеристик, рейтингів користувачів та відгуків.

Вебзастосунки з продажу автомобілів також сприяють збільшенню ефективності для автодилерів і приватних продавців. Вони надають можливість швидко розміщувати оголошення про продаж автомобілів, взаємодіяти з потенційними покупцями і вести електронну облікову запису про угоди. Це сприяє зменшенню затрат на маркетинг та рекламу, а також прискорює процес продажу.

Проте, вебзастосунки з продажу автомобілів також постають перед деякими викликами. Найбільш значимими з них є безпека та довіра до онлайн-платформ, оскільки покупці повинні передавати особисту та фінансову інформацію. Додатково, необхідно забезпечувати якісну технічну підтримку для користувачів та забезпечити точність та актуальність інформації про автомобілі.

У загальному, вебзастосунки з продажу автомобілів мають великий потенціал у поліпшенні процесу купівлі та продажу автомобілів. Завдяки їхній зручності, доступності та широкому функціоналу вони можуть сприяти розвитку автомобільного ринку і задоволенню потреб різних груп покупців та продавців.

Після аналізу різних технологій розробки була обрана оптимальна платформа веброзробки та база даних, які найкраще відповідають вимогам проекту. З використанням цих технологій було успішно розроблено вебзастосунок.

При розробці макету сайту було приділено особливу увагу створенню зручного та інтуїтивно зрозумілого інтерфейсу для користувачів. Це дало змогу створити зручну та ефективну систему надання соціальних послуг та отримання необхідної інформації. Розроблений макет сайту має відповідний дизайн, логічну структуру та інтуїтивно зрозумілі розділи, що сприяє легкій навігації та використанню функціоналу вебзастосунку.

У результаті використання обраних технологій та створення зручного інтерфейсу, розроблений вебзастосунок став ефективним інструментом для користувачів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. JavaScript history. URL: https://www.w3schools.com/js/js_history.asp (дата звернення 24.05.2023).
2. Webdesigner Depot. URL: <https://www.webdesignerdepot.com/> (дата звернення 24.05.2023).
3. Smashing Magazine. URL: <https://www.smashingmagazine.com/> (дата звернення 24.05.2023).
4. Learn to Code HTML & CSS. URL: <https://learn.shayhowe.com/html-css/> (дата звернення 24.05.2023).
5. Офіційна документація PHP. URL: <https://www.php.net/manual/en/> (дата звернення 24.05.2023).
6. PHP: The Right Way – комплексний гайд по розробці на PHP. URL: <https://phprightway.com/> (дата звернення 24.05.2023).
7. Акопян С. Як написати і продати свій PHP скрипт. 2021. 86 с.
8. MySQL документація. URL: <https://dev.mysql.com/doc/> (дата звернення 24.05.2023).
9. Що таке use case та для чого вони потрібні. URL: <https://training.qatestlab.com/blog/technical-articles/what-is-a-use-case-and-what-are-they-for/> (дата звернення 24.05.2023).
10. Офіційний сайт Symfony. URL: <https://symfony.com/> (дата звернення: 26.04.2023).
11. Miller B. D. Principles of web design. Allworth, 2022. 280 p.
12. Hong P. Practical web design: Learn the fundamentals of web design with HTML5, CSS3, Bootstrap, JQuery, and Vue.js. Packt Publ., 2018. 368 p.
13. Dorst K. Notes on design: How creative practice works. BIS Publishers, 2018. 208 p.
14. Wasserman E. A. As if by design: How creative behaviors really evolve. Cambridge University Press, 2021. 328 p.
15. Структура сайту: основні види та правила їх розробки. URL:

<https://webtune.com.ua/statti/web-rozrobka/struktura-sajtu/> (дата звернення 24.05.2023).

16. Livechat. URL: <https://www.livechat.com/ru/> (дата звернення 24.05.2023).

17. Huang W., Yang P. Application analysis and research of artificial intelligence technology in the creative stage of web design. *BCP Social Sciences & Humanities*. 2022. Vol. 20. P.535–543. DOI: 10.54691/bcpssh.v20i.2389.

18. Metanit. URL: <https://metanit.com/web/html5/5.2.php> (дата звернення 10.06.2023).

19. W3Schools. URL: https://www.w3schools.com/php/php_mysql_select.asp (дата звернення 10.06.2023).

20. JavaScript.ru. URL: <https://learn.javascript.ru/alert-prompt-confirm> (дата звернення 10.06.2023).

ДОДАТОК А

Код програми

Index.PHP

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Car Shop</title>
  <link rel="stylesheet" href="styles/main.css">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">
</head>
<body>
  </head>
<body>
  <header>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
      <div class="container">
        <a class="navbar-brand" href="index.php">Автокранниця</a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-
controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
          <ul class="navbar-nav ml-auto">
            <li class="nav-item">
              <a class="nav-link" href="salout.php">Нові пропозиції</a>
            </li>
            <?php
              if($_COOKIE['user_id'] == ") :
            <?>
            <li class="nav-item">
              <a class="nav-link" href="reg.html">Зареєструватись</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="auth.html">Авторизація</a>
            </li>
            <?php else: ?>
            <li class="nav-item">
              <a class="nav-link" href="announcement.html">Створити оголошення</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="admin.html">Особистий кабінет</a>
            </li>

```

```

    <li class="nav-item">
      <a class="nav-link" href="exit.php">Вийти</a>
    </li>
  <?php endif; ?>
</ul>
</div>
</div>
</nav>
</header>

<main>
  <div class="bg" style="background-image: url(img/1640585509_19-damion-club-p-mashina-v-gorakh-priroda-23.jpg);">

  <section class="banner">

    <h2>Вітаємо в автомагазині</h2>
    <p>Знайди авто своєї мрії за найкращою ціною</p>
    <a href="#" class="search">Пошук</a>
  </section>
</div>

  <section class="car-listing">

    <div class="container">

      <div class="row">
        <div class="col-lg-4 col-md-4 col-sm-6">

          <div class="car">
            
            <h3>Opel Grandland (2023)</h3>
            <p>Ціна: $26,000</p>
            <button class="view-details">Переглянути</button>
          </div>

        </div>

        <div class="modal">
          <div class="modal-content">
            <span class="close">&times;</span>
            
            <h3 class="product-title">Название автомобиля</h3>
            <p class="product-description">Описание автомобиля</p>
            <p class="product-price">Цена: $10000</p>
            <button class="add-to-cart">Добавить в корзину</button>
          </div>
        </div>
      </div>
    </div>
  </section>

```



```

<script src="js/itemscript.js"></script>

<div class="col-lg-4 col-md-4 col-sm-6">

<div class="car">
  
  <h3>Toyota Camry (2015)</h3>
  <p>Ціна: $15,000</p>
  <button class="view-details">Переглянути</button>
</div>
</div>
<div class="modal">
  <div class="modal-content">
    <span class="close">&times;</span>
    
    <h3 class="product-title">Название автомобиля</h3>
    <p class="product-description">Описание автомобиля</p>
    <p class="product-price">Цена: $10000</p>
    <button class="add-to-cart">Добавить в корзину</button>
  </div>
</div>
<script src="js/itemscript.js"></script>

<div class="col-lg-4 col-md-4 col-sm-6">

<div class="car">
  
  <h3>Hyundai Tucson (2018)</h3>
  <p>Ціна: $20,000</p>
  <button class="view-details">Переглянути</button>
</div>

</div>

<div class="modal">
  <div class="modal-content">
    <span class="close">&times;</span>
    
    <h3 class="product-title">Название автомобиля</h3>
    <p class="product-description">Описание автомобиля</p>
    <p class="product-price">Цена: $10000</p>
    <button class="add-to-cart">Добавить в корзину</button>
  </div>
</div>
<script src="js/itemscript.js"></script>
</div>

```

```

    </div>
  </section>

</main>

<footer>
  <ul>
    <li><a href="redactt.php">Про нас</a></li>
    <li><a href="123.html">Правила спільноти</a></li>
    <li><a href="#">Служба підтримки</a></li>
    <li><a href="#">FAQ</a></li>
  </ul>
</footer>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"></script>
</body>

</html>

```

reg.Html

```

<!DOCTYPE html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Registration</title>
  <link rel="stylesheet" href="styles/style.css">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">
</head>
<body style="background-image: url(img/1640585509_19-damion-club-p-mashina-v-gorakh-priroda-23.jpg);">
  <header>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
      <a class="navbar-brand" href="index.php">Автокрамниця</a>
    </nav>
  </header>
  <main>
    <div class="container">
      <section class="registration-form">
        <h2>Регістрація</h2>
        <form action="reg.php" method="post">
          <div class="form-group">
            <label for="name">ім`я:</label>
            <input type="text" id="name" name="name" class="form" placeholder="введіть ваше ім`я">
          </div>
          <div class="form-group">
            <label for="email">Email:</label>
            <input type="email" id="email" name="email" class="form" placeholder="введіть ваш Email">
          </div>
          <div class="form-group">
            <label for="password">Пароль:</label>
            <input type="password" id="password" name="password" class="form" placeholder="придумайте пароль">
          </div>
          <div class="form-group">
            <label for="confirm-password">Підтвердіть пароль:</label>
            <input type="password" id="confirm-password" name="confirm-password" class="form"
            placeholder="повторіть пароль">
          </div>
          <button class="btn btn-dark col-lg-12 col-md-12 col-sm-12" type="submit">Зареєструватись</button>
        </form>
      </section>
    </div>
  </main>

```

```

    </form>
  </section>
</div>
</main>
</body>
</html>

```

reg.PHP

```

<?php
$name = filter_var(trim($_POST['name']),
);

$email = filter_var(trim($_POST['email']),
);

$password = filter_var(trim($_POST['password']),
);

$pass = filter_var(trim($_POST['confirm-password']),
);

if(mb_strlen($name) < 2 || mb_strlen($name) > 15){
echo "дерьмо";
exit();
}

$password= md5($password. "fhdjeudj456f");
$pass= md5($pass. "fhdjeudj456f");

$mysql = new mysqli('localhost','root','root','reg-db');
$mysql->query("INSERT INTO `user` (`email`,`pass`,`pass2`,`name`)
VALUES('$email','$password','$pass','$name')");

$mysql ->close();
header('Location: index.php');
?>

```

auth.PHP

```

<?php
$email = filter_var(trim($_POST['email']),FILTER_SANITIZE_EMAIL
);

$password = filter_var(trim($_POST['password']),FILTER_SANITIZE_STRING
);

$password= md5($password. "fhdjeudj456f");

$mysql = new mysqli('localhost','root','root','reg-db');

$result = $mysql->query("SELECT * FROM `user` WHERE `email` =
'$email' AND `pass` = '$password'");

```

```

$masive = $result->fetch_assoc();
if(count($masive) == 0){

    }

setcookie('user_id', $masive['id'], time() + 3600, "/");

$mysql ->close();

header('Location: index.php');
?>

```

announcement.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Добавить объявление</title>
    <link rel="stylesheet" href="styles/announcement.css">
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
</head>
<body style="background-image: url(img/1640585509_19-damion-club-p-mashina-v-gorakh-priroda-23.jpg); ;">

    <header>
        <nav class="navbar navbar-expand-lg navbar-light bg-light">
            <a class="navbar-brand" href="index.php">Автокранниця</a>
        </header>

    <div class="container" >
        <h1 class="mt-5">Додати оголошення</h1>
        <form action="sale.php" method="POST" enctype="multipart/form-data">
            <div class="form-group">
                <label for="make">Марка автомобіля:</label>
                <input type="text" class="form-control" id="make" name="make" required>
            </div>

            <div class="form-group">
                <label for="model">Модель автомобіля:</label>
                <input type="text" class="form-control" id="model" name="model" required>
            </div>

```

```
</div>

<div class="form-group">
  <label for="price">Ціна:</label>
  <input type="number" class="form-control" id="price" name="price" required>
</div>

<div class="form-group">
  <label for="description">Опис:</label>
  <textarea class="form-control" id="description" name="description" required></textarea>
</div>

<div class="form-group">
  <label for="photo">Фотографії:</label>
  <input type="file" class="form-control-file" id="photo" name="photo[]" multiple>
</div>

<div class="form-group">
  <label for="year">Рік:</label>
  <input type="number" class="form-control" id="year" name="year" required>
</div>

<div class="form-group">
  <label for="region">Регіон:</label>
  <input type="text" class="form-control" id="region" name="region" required>
</div>

<button type="submit" class="btn btn-dark">Додати оголошення</button>
</form>
</div>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>
```

announcement.PHP

```
<?php
$servername = "localhost";
$username = "root";
$password = "root";
$dbname = "reg-db";
```

```
$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Ошибка подключения: " . $conn->connect_error);
}

$userId = $_COOKIE['user_id'];

$make = $_POST['make'];
$model = $_POST['model'];
$price = $_POST['price'];
$description = $_POST['description'];

$targetDir = "uploads/";
$uploadedPhotos = [];

foreach ($_FILES['photo']['tmp_name'] as $index => $tmpName) {
    $targetFile = $targetDir . basename($_FILES['photo']['name'][$index]);
    move_uploaded_file($tmpName, $targetFile);
    $uploadedPhotos[] = $targetFile;
}

$sql = "INSERT INTO carsale (make, model, price, description, image, user_id)
VALUES ('$make', '$model', '$price', '$description', " . implode(", ", $uploadedPhotos) . ", '$userId')";

if ($conn->query($sql) === TRUE) {
    header('Location: admin.php');
} else {
    echo "Ошибка при добавлении объявления: " . $conn->error;
}

$conn->close();
?>
```

Announcementlist.PHP

```
<?php
$servername = "localhost";
$username = "root";
$password = "root";
$dbname = "reg-db";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Ошибка подключения: " . $conn->connect_error);
}

$sql = "SELECT * FROM carsale";
$result = $conn->query($sql);

if ($result->num_rows > 0) {

    while ($row = $result->fetch_assoc()) {
        $image = explode(",", $row['image']);
        $make = $row['make'];
        $model = $row['model'];
        $price = $row['price'];
        $description = $row['description'];
        $year = $row['year'];
        $region = $row['region'];

        echo "<div class='listing'>";

        echo "<h2>$make $model</h2>";

        echo "<div class='photos'>";
        foreach ($image as $img) {
            echo "<img src='$img' alt='Car Photo' class='img-fluid'>";
        }
        echo "</div>";
        echo "<p>Ціна $: $price</p>";
    }
}
```

```

echo "<p>Опис: $description</p>";
echo "<p>Рік: $year</p>";
echo "<p>Регіон:$region</p>";
echo "</div>";
}
} else {
    echo "Нет доступных объявлений.";
}

```

```

$conn->close();
?>

```

Userpage.PHP

```

<?php

$servername = "localhost";
$username = "root";
$password = "root";
$dbname = "reg-db";
$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Ошибка подключения: " . $conn->connect_error);
}

session_start();
if (!isset($_SESSION['user']) && !isset($_COOKIE['user_id'])) {
    header("Location: auth.html"); // Перенаправление на страницу авторизации, если пользователь не авторизован
    exit();
}

$userId = isset($_SESSION['user']['id']) ? $_SESSION['user']['id'] : $_COOKIE['user_id'];

$sql = "SELECT * FROM carsale WHERE user_id = $userId";
$result = $conn->query($sql);

```



```
if ($result->num_rows > 0) {  
  
    while ($row = $result->fetch_assoc()) {  
        $make = $row['make'];  
        $model = $row['model'];  
        $price = $row['price'];  
        $description = $row['description'];  
        $year = $row['year'];  
        $region = $row['region'];  
        $image = explode(",", $row['image']);  
  
        echo "<div class='listing'>";  
        echo "<h2>$make $model</h2>";  
        echo "<div class='photos'>";  
        foreach ($image as $img) {  
            echo "<img src='$img' alt='Car Photo' class='img-fluid'>";  
        }  
        echo "</div>";  
        echo "<p>Цена: $price</p>";  
        echo "<p>Описание: $description</p>";  
        echo "<p>Рік: $year</p>";  
        echo "<p>Регіон: $region</p>";  
        echo "</div>";  
    }  
} else {  
  
}  
  
$conn->close();  
?>
```