**МIНIСТЕРСТВО ОСВIТИ І НАУКИ**

**НАЦІОНАЛЬНИЙ ТРАНСПОРТНИЙ УНІВЕРСИТЕТ Кафедра Інформаційних систем і технологій**

**КУРСОВА РОБОТА**

**з дисципліни ««КРОС-ПЛАТФОРМНІ ТЕХНОЛОГІЇ РОЗРОБКИ ПЗ»» на тему**

**«Додаток системи обліку продажу автомобілів на вторинному ринку»**

**Виконала студентка: Курінна Даша**

**Група: ІБК-4-1**

**Перевірив:**

**Київ 2023**

**ЗМІСТ**

[Вступ 3](#_Toc150035026)

[Розділ 1. Веб програмування 4](#_Toc150035027)

[Розділ 2. Розробка веб додатків 7](#_Toc150035028)

[Розділ 3. Реалізація веб додатка 11](#_Toc150035029)

[2.1 Дизайн додатку 11](#_Toc150035030)

[2.2 Демонстрація функціоналу 14](#_Toc150035031)

[Висновок 16](#_Toc150035032)

[Використані джерела 17](#_Toc150035033)

[Додатки 18](#_Toc150035034)

# Вступ

Веб програмування - це процес створення програмного забезпечення, яке працює в Інтернеті. Веб-програми, як правило, складаються з двох частин: клієнтської частини, яка працює на стороні користувача, і серверної частини, яка працює на сервері.

Для веб-програмування використовується ряд мов програмування, зокрема HTML, CSS, JavaScript, PHP, Python та Java. HTML використовується для створення структури веб-сторінки, CSS - для її стилізації, а JavaScript - для додавання інтерактивності. PHP, Python та Java використовуються для розробки серверної частини веб-програм.

HTML (HyperText Markup Language) - це мова розмітки гіпертексту, яка використовується для створення структури веб-сторінок. HTML-код складається з тегів, які вказують браузеру, як відображати вміст сторінки.

HTML був створений британським вченим Тимом Бернерсом-Лі в 1989 році. Перша версія HTML, HTML 1.0, була опублікована в 1993 році. З тих пір було випущено кілька версій HTML, включаючи HTML 5, яка є найсучаснішою версією.

CSS (Cascading Style Sheets) - це мова стилізації, яка використовується для форматування веб-сторінок. CSS-код використовується для зміни шрифтів, кольорів, розмірів і розташування елементів на веб-сторінці.

CSS був розроблений в 1995 році Брендіном Айхом, співробітником Netscape Communications. Перша версія CSS, CSS 1, була опублікована в 1996 році. З тих пір було випущено кілька версій CSS, включаючи CSS 3, яка є найсучаснішою версією.

JavaScript - це мова програмування, яка використовується для додавання інтерактивності веб-сторінкам. JavaScript-код використовується для створення таких елементів, як випадаючі списки, кнопки і вікна.

JavaScript був розроблений Brendan Eich в 1995 році, коли він працював в Netscape Communications. Перша версія JavaScript, LiveScript, була опублікована в 1995 році. З тих пір було випущено кілька версій JavaScript, включаючи ECMAScript 2023, яка є найсучаснішою версією.

HTML, CSS і JavaScript є трьома основними мовами програмування, які використовуються для створення веб-сайтів і веб-додатків. Ці мови працюють разом, щоб створити веб-сторінки, які є як структурованими, так і стильними, а також інтерактивними.

# Розділ 1. Веб програмування

**Веб програмування** - це процес створення програмного забезпечення, яке працює на веб-серверах і надає можливість взаємодії з користувачами через веб-браузери. Веб програмування включає розробку веб-додатків та веб-сайтів і охоплює всі аспекти їх створення, від проектування і структури до розробки, тестування та впровадження.

Основні компоненти веб програмування включають:

1. **HTML (Hypertext Markup Language)**: HTML визначає структуру веб-сторінки та розміщення текстового та мультимедійного контенту. Він використовується для створення скелету веб-сторінок.
2. **CSS (Cascading Style Sheets)**: CSS відповідає за зовнішній вигляд веб-сторінок, включаючи кольори, шрифти, розміщення та інші аспекти дизайну.
3. **JavaScript**: JavaScript - це мова програмування, яка використовується для надання інтерактивності веб-сторінкам. Вона дозволяє створювати реактивні елементи, взаємодіяти з користувачем і сервером, а також обробляти події на сторінці.
4. **Бекенд**: Бекенд включає серверну частину веб-додатка, яка обробляє запити від клієнтів, взаємодіє з базами даних і надає дані клієнтам. Бекенд може бути написаний на різних мовах програмування, таких як Python, Ruby, Java, Node.js і інші.
5. **Бази даних**: Бази даних використовуються для збереження та управління даними, які використовуються в веб-додатках. Вони дозволяють зберігати інформацію, таку як дані користувачів, продукти, замовлення тощо.
6. **Сервери**: Сервери відповідають за розміщення веб-додатків та надання доступу до них через Інтернет. Вони приймають запити від клієнтів, обробляють їх і надсилають відповіді.

Веб програмування має безліч переваг, які роблять його привабливим для розробників, користувачів і бізнесу. Ось деякі з основних переваг веб програмування:

* **Доступність**: Веб-додатки доступні з будь-якого місця з доступом до Інтернету, що робить їх зручними для користувачів. Користувачам не потрібно встановлювати спеціальне програмне забезпечення, вони можуть просто відкрити браузер і перейти на веб-сторінку.
* **Крос-платформеність**: Веб-додатки працюють на різних операційних системах (Windows, macOS, Linux) і пристроях (комп'ютери, смартфони, планшети). Розробникам не потрібно створювати окремі версії для кожної платформи.
* **Оновлення**: Оновлення веб-додатків відбуваються на сервері, тому користувачам не потрібно вручну встановлювати оновлення. Це дозволяє розробникам швидко виправляти помилки і додавати новий функціонал.
* **Спрощене розгортання**: Розгортання веб-додатків на серверах відбувається легко і швидко. Розробники можуть використовувати хмарні рішення або власні сервери.
* **Легкий доступ до бази даних**: Веб-додатки можуть взаємодіяти з базами даних для зберігання та управління інформацією. Це дозволяє створювати додатки, які працюють з великим обсягом даних.
* **Спільна розробка**: Веб-додатки дозволяють багатьом розробникам працювати над проектом з різних місць і об'єднувати свої зусилля за допомогою систем контролю версій.
* **Споживачам не потрібен спеціальний софт**: Веб-додатки відкриваються в стандартних веб-браузерах, тому користувачам не потрібно встановлювати додаткове програмне забезпечення або додатки.
* **Швидкий розповсюдження і оновлення**: Веб-додатки можуть бути швидко оновлені та розповсюджені серед користувачів, що полегшує впровадження нових функцій або виправлення помилок.
* **Інтеграція з іншими сервісами**: Веб-додатки легко інтегруються з іншими сервісами і додатками, що дозволяє створювати розширені функціональність та використовувати зовнішні API.
* **Аналітика і відстеження**: Веб-додатки можуть включати інструменти для збору аналітики та відстеження поведінки користувачів, що допомагає аналізувати діяльність і вдосконалювати додаток.

Ці переваги роблять веб програмування привабливим і важливим аспектом розвитку сучасних програмних продуктів і сервісів.

У сучасному веб-програмуванні існує безліч фреймворків, які спрощують процес створення веб-додатків і дозволяють розробникам прискорити роботу. Вони надають готові інструменти та структуру для створення веб-додатків швидше і ефективніше. Проте, у нашому проекті ми вирішили не використовувати фреймворки з метою простоти та більшого контролю над розробкою.

Ось декілька прикладів популярних фреймворків для веб розробки:

1. **React**: React - це бібліотека JavaScript для створення інтерфейсів користувача, яка використовується для розробки односторінкових додатків (SPA). Він відомий своєю декларативною синтаксисом і високою продуктивністю.
2. **Angular**: Angular - це фреймворк JavaScript, розроблений компанією Google, який дозволяє створювати складні веб-додатки. Він надає інструменти для роботи з компонентами, маршрутизацією, HTTP-запитами та іншими функціями.
3. **Vue.js**: Vue.js - це інша бібліотека JavaScript для створення інтерфейсів користувача. Він славиться своєю легкістю вивчення і використання, а також можливістю інтеграції з іншими проектами.
4. **Express.js**: Express.js - це мінімалістичний фреймворк для створення серверної частини веб-додатків на платформі Node.js. Він дозволяє легко створювати API та обробляти HTTP-запити.
5. **Ruby on Rails**: Ruby on Rails - це фреймворк для розробки веб-додатків, який використовує мову програмування Ruby. Він надає ряд зручних інструментів для створення баз даних, обробки запитів і створення веб-сайтів.

Хоча ці фреймворки допомагають розробникам ефективніше створювати веб-додатки, у нашому проекті ми вирішили обмежитися чистими технологіями HTML, CSS і JavaScript без використання фреймворків. Це дозволить нам краще розуміти основи веб-програмування і мати більший контроль над кожним аспектом розробки.

# Розділ 2. Розробка веб додатків

Процес розробки веб-додатків - це послідовність кроків, які розробник виконує для створення функціонального і надійного веб-додатка. Цей процес включає в себе кілька ключових етапів, від планування і проектування до розробки, тестування та випуску додатка. Ось загальна структура процесу розробки веб-додатків:

1. **Планування і аналіз**:
   * Визначення мети і цільової аудиторії додатка.
   * Вивчення конкурентів і аналіз ринку.
   * Визначення функціональних вимог та функцій додатка.
2. **Проектування і інтерфейс користувача**:
   * Розробка схеми бази даних та структури додатка.
   * Створення макетів і дизайну інтерфейсу користувача (UI).
   * Вибір технологій і стеку розробки.
3. **Розробка**:
   * Створення фронтенду додатка, включаючи розмітку (HTML), стилі (CSS) і функціональність (JavaScript).
   * Розробка бекенду для обробки запитів, взаємодії з базою даних та надання даних на фронтенд.
   * Інтеграція фронтенду і бекенду.
4. **Тестування**:
   * Виконання різних видів тестів, включаючи тестування функціональності, юзабільності і безпеки.
   * Виправлення помилок і недоліків.
5. **Оптимізація та покращення**:
   * Виправлення і покращення продуктивності і швидкості додатка.
   * Оптимізація коду та використання кешування.
   * Додавання нового функціоналу та покращення існуючого.
6. **Випуск та розгортання**:
   * Підготовка додатка до випуску.
   * Розгортання на веб-сервері або іншій платформі.
   * Налаштування моніторингу та журналювання.
7. **Підтримка та післявипускні роботи**:
   * Постійна підтримка і оновлення додатка.
   * Вирішення проблем та недоліків після випуску.
   * Взаємодія з користувачами та врахування їхніх відгуків.

Це загальна структура процесу розробки веб-додатків, яка може бути адаптована до конкретних потреб і вимог проекту.

Розробка функціональності веб-додатка є однією з ключових частин процесу створення веб-додатка на вторинному ринку автомобілів. Цей етап включає в себе впровадження всіх необхідних функцій та можливостей, які дозволять користувачам зручно взаємодіяти з додатком і здійснювати пошук та перегляд автомобілів. Основні функції додатку включають:

* **Пошук автомобілів**: Розробка можливості пошуку автомобілів за різними критеріями, такими як марка, модель, рік випуску, ціна, пробіг, тип трансмісії тощо.
* **Список автомобілів**: Створення сторінки зі списком доступних автомобілів, яка включає інформацію про кожен автомобіль, таку як фотографія, марка, модель, ціна, рік випуску, тип двигуна, пробіг, місто та тип трансмісії.
* **Фільтри та сортування**: Розробка можливості фільтрації автомобілів за різними параметрами, а також сортування результатів пошуку.
* **Детальна інформація про автомобіль**: Створення сторінки з докладною інформацією про кожен автомобіль, включаючи фотографії, технічні характеристики, опис і контактну інформацію продавця.
* **Додавання автомобілів**: Розробка можливості додавання нових автомобілів до бази даних додатку продавцями.
* **Зберігання та обробка даних**: Розробка механізмів для зберігання та обробки даних про автомобілі, користувачів та іншу інформацію.
* **Пошук за допомогою фільтрів та сортуванням**: Розробка можливості фільтрації автомобілів за різними параметрами та сортування результатів.
* **Оптимізація швидкості завантаження сторінок**: Розробка для досягнення високої продуктивності та швидкості завантаження сторінок.

Розробка веб-додатків може виконуватися на різних платформах та середовищах, які надають різноманітні можливості для створення інтерактивних інтерфейсів та функціональних додатків. Деякі з популярних платформ і середовищ включають:

1. **Visual Studio Code**: Це безкоштовна розробницька середовище, яке підтримує багато мов програмування і фреймворків. Visual Studio Code має широкий вибір розширень, які полегшують роботу розробників веб-додатків.
2. **Sublime Text**: Ще один популярний текстовий редактор, який використовується для розробки веб-додатків. Він відомий своєю швидкістю та легкістю використання.
3. **WebStorm**: Це інтегроване середовище розробки, призначене спеціально для веб-розробки на JavaScript. WebStorm надає широкий спектр інструментів для розробки та налагодження веб-додатків.
4. **Eclipse**: Якщо ви розробляєте веб-додатки на Java, то Eclipse може бути вашим вибором. Він надає підтримку для розробки на Java та інших мовах програмування.
5. **IntelliJ IDEA**: Це інша платформа для розробки на Java, яка надає потужні інструменти для створення веб-додатків.
6. **NetBeans**: Якщо вам потрібно розробляти веб-додатки на різних мовах програмування, NetBeans може бути варіантом, оскільки він підтримує багато мов програмування та фреймворків.
7. **Atom**: Atom - це інший безкоштовний текстовий редактор, розроблений компанією GitHub. Він підтримує розширення та має активну спільноту розробників.
8. **Brackets**: Це безкоштовне середовище розробки, спеціально призначене для роботи з веб-технологіями, такими як HTML, CSS та JavaScript.
9. **CodePen**: CodePen - це онлайн-середовище для розробки веб-додатків, яке дозволяє створювати, тестувати та ділитися кодом та демонстраціями.
10. **GitHub Codespaces**: Це інтерактивне середовище для розробки веб-додатків в хмарі, яке інтегроване з GitHub та дозволяє розробникам спільно працювати над проектами.

Обираючи платформу для розробки веб-додатків, важливо враховувати свої потреби та завдання проекту, а також власні зручності та досвід.

# Розділ 3. Реалізація веб додатка

## 2.1 Дизайн додатку

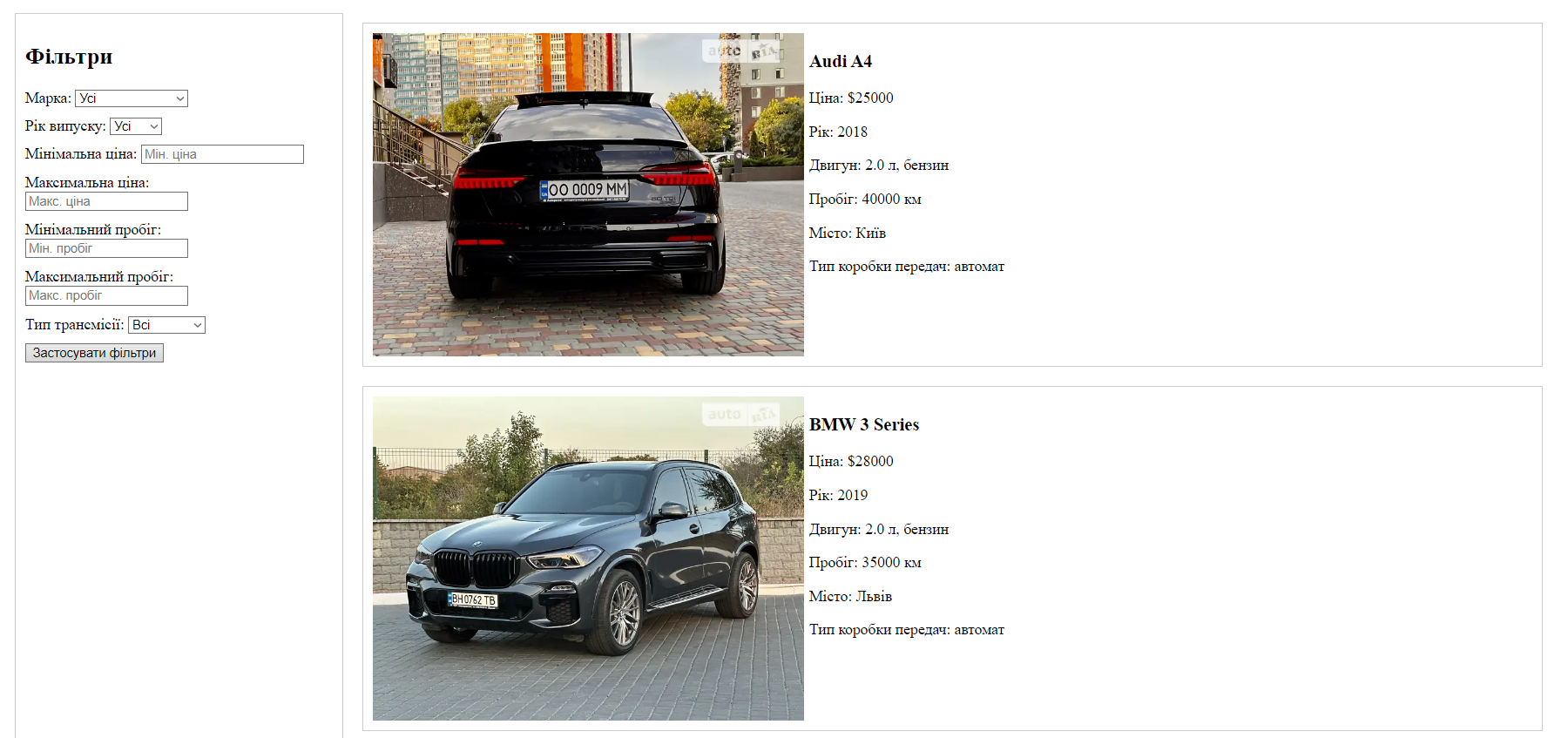
****

Рис 2.1.1 – Головний екран

Після запуску програми, користувач бачить головну сторінку сайту.

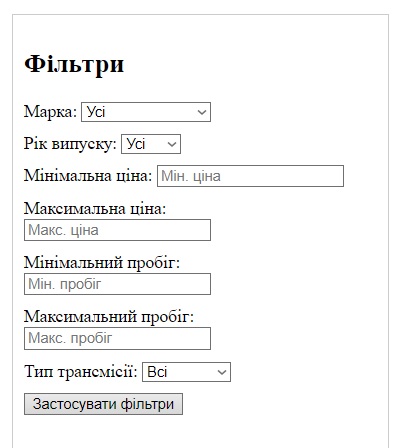


Рис 2.1.2 – Меню вибору фільтрів

Меню вибору фільтрів є важливою складовою вашого веб-додатку, оскільки воно дозволяє користувачам точно вибирати автомобілі з великого списку за різними параметрами. Нижче наведено опис, як може виглядати це меню:

* **Бренд (Марка)**: У меню бренда користувачі можуть вибрати конкретний автомобільний бренд. Можливість вибору зі списку брендів (наприклад, Audi, BMW, Mercedes-Benz) або введення бренда вручну.
* **Рік випуску**: Фільтр "Рік випуску" дозволяє користувачам вибирати автомобілі за роком випуску. Можливість вказати мінімальний та максимальний рік випуску.
* **Ціна**: Користувачі можуть вибирати автомобілі за ціною. Введення мінімальної та максимальної ціни або вибір цінового діапазону.
* **Пробіг**: Можливість вибору автомобілів за пробігом. Користувачі можуть вказати мінімальний та максимальний пробіг.
* **Тип коробки передач**: Можливість вибору автомобілів з автоматичною або механічною коробкою передач.

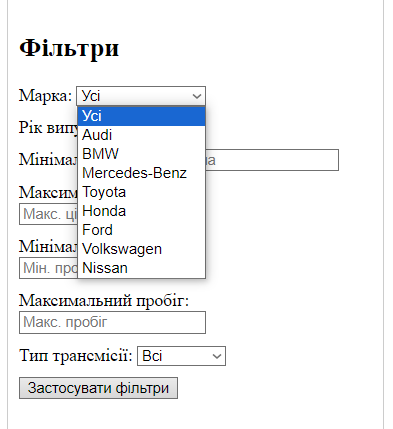


Рис 2.1.3 – Меню вибору марки машини

Вибір марки машини та року випуску мають особливий функціонал оскільки вибірка марок підв’язана до бази даних, якщо ми додамо в базу даних якусь нову марку то вона автоматично з’явится і в меню вибору, аналогічний функціонал має меню вибору року випуску.

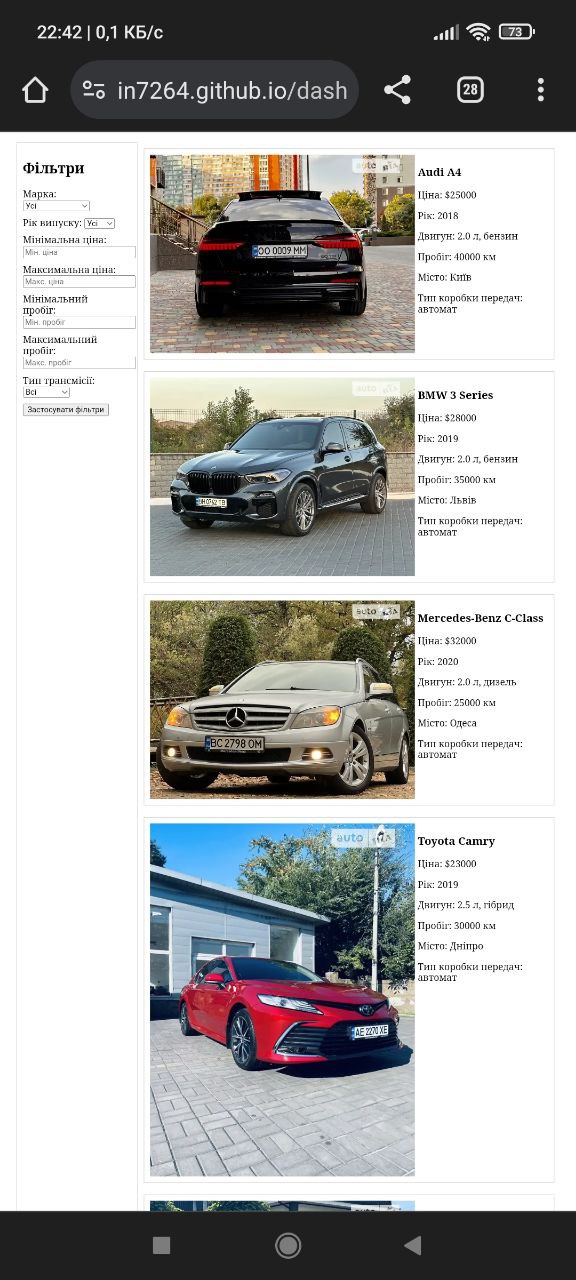


Рис 2.1.4 – Вигляд програми адаптований для мобільних пристроїв

Адаптація програми для мобільних пристроїв є важливою характеристикою веб-додатку. Це означає, що додаток може зручно використовуватися на різних типах мобільних пристроїв, таких як смартфони і планшети. Веб-додаток має респонсивний дизайн, який адаптується до різних розмірів екранів, включаючи мобільні пристрої, планшети та десктопи. Це означає, що елементи і контент автоматично масштабуються та переструктуруються, щоб оптимізувати відображення на маленьких екранах.

## 2.2 Демонстрація функціоналу

Наприклад я хочу знайти знайти Audi 2018 року випуску з ціновим діапазоном більше 22000$ але менше 30000$ та з максимальним пробігом в 100000. Результат на ПК версії:

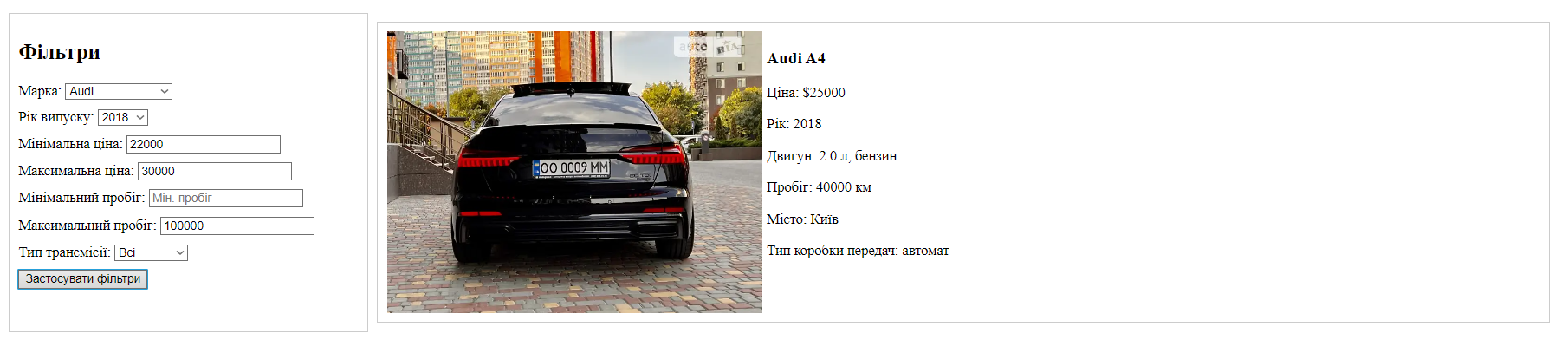


Рис 2.2.1 – Вигляд програми адаптований для мобільних пристроїв

Результат на телефонній версії:

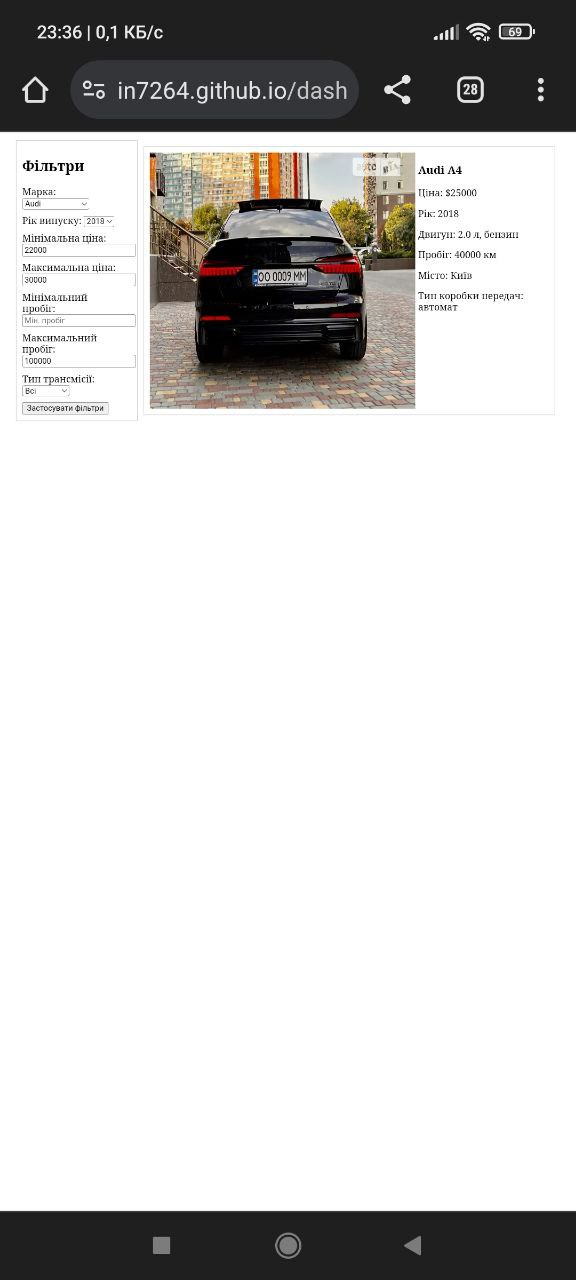


Рис 2.2.2 – Вигляд програми адаптований для мобільних пристроїв

# Висновок

У цій курсовій роботі була розглянута розробка веб-додатку для системи обліку продажу автомобілів на вторинному ринку. Розробка веб-додатку є актуальною завданням, оскільки вона відповідає потребам користувачів у пошуку та продажу автомобілів. Даний веб-додаток надає можливість користувачам швидко та зручно знаходити автомобілі за різними параметрами, такими як марка, модель, рік випуску, ціна, пробіг та інші.

У роботі було розглянуто такі ключові аспекти:

1. **Веб програмування та розробка веб-додатків**: Було розглянуто принципи та технології веб програмування, а також процес розробки веб-додатків. Зокрема, були описані інструменти та платформи, які можна використовувати для розробки веб-додатків.
2. **Розробка функціональності додатку**: Веб-додаток було розроблено з урахуванням функціональних вимог, включаючи фільтрацію автомобілів за різними критеріями, відображення деталей кожного автомобіля та можливість редагування та додавання нових автомобілів.
3. **Адаптація для мобільних пристроїв**: Важливою особливістю цього веб-додатку є його адаптація для мобільних пристроїв. Респонсивний дизайн та оптимізація завантаження роблять його доступним для користувачів на різних пристроях, включаючи смартфони та планшети.

Завдяки цій курсовій роботі, було розроблено веб-додаток, який спрощує процес обліку продажу автомобілів на вторинному ринку. Він надає користувачам інструменти для швидкого та зручного пошуку та продажу автомобілів. Розробка цього додатку сприяє покращенню доступності інформації та допомагає спростити процес покупки та продажу автомобілів для користувачів.

# Використані джерела

1. Веб програмування. Підручник. Автор: М. В. Гаврилюк. Видавництво: "Вища школа", Київ, 2022.
2. Кейт Джонс. DOM Scripting: Web Design with JavaScript and the Document Object Model. / К. Джонс — Перше, 2005. — 368 c
3. Веб-розробка. Практичний курс. Автор: С. М. Шевчук. Видавництво: "Софтпрес", Київ, 2021.
4. Методи написання web-сайтів [Електронний ресурс]. - Режим доступу http://webstudio2u.net/ru/webdesign/354-site-developmethods.html
5. Веб-технології. Підручник. Автор: В. П. Лавров. Видавництво: "Київський національний університет імені Тараса Шевченка", Київ, 2020.
6. Стаття на Wikipedia про HTML5 [Електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/HTML5>
7. Стаття про CSS3 [Електронний ресурс]. – Режим доступу: http://htmlbook.ua/css3
8. Веб-розробка. Посібник. Автор: Є. О. Волкова. Видавництво: "Вища школа", Київ, 2021.

# Додатки

html:

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <title>Система обліку продажу автомобілів</title>

    <link rel="stylesheet" type="text/css" href="style.css">

</head>

<body>

    <div class="container">

        <div class="filters">

            <h2>Фільтри</h2>

            <div class="filter">

                <label for="brandFilter">Марка:</label>

                <select id="brandFilter">

                    <option value="all">Усі</option>

                </select>

            </div>

            <div class="filter">

                <label for="yearFilter">Рік випуску:</label>

                <select id="yearFilter">

                    <option value="all">Усі</option>

                </select>

            </div>

            <div class="filter">

                <label for="priceFilterMin">Мінімальна ціна:</label>

                <input type="number" id="priceFilterMin" placeholder="Мін. ціна">

            </div>

            <div class="filter">

                <label for="priceFilterMax">Максимальна ціна:</label>

                <input type="number" id="priceFilterMax" placeholder="Макс. ціна">

            </div>

            <div class="filter">

                <label for="mileageFilterMin">Мінімальний пробіг:</label>

                <input type="number" id="mileageFilterMin" placeholder="Мін. пробіг">

            </div>

            <div class="filter">

                <label for="mileageFilterMax">Максимальний пробіг:</label>

                <input type="number" id="mileageFilterMax" placeholder="Макс. пробіг">

            </div>

            <div class="filter">

                <label for="transmissionFilter">Тип трансмісії:</label>

                <select id="transmissionFilter">

                    <option value="all">Всі</option>

                    <option value="автомат">Автомат</option>

                    <option value="механіка">Механіка</option>

                </select>

            </div>

            <button id="applyFilters">Застосувати фільтри</button>

        </div>

        <div class="car-list">

            <!--список автомобілів -->

        </div>

    </div>

    <script src="script.js"></script>

</body>

</html>

css:

.container {

    display: flex;

    justify-content: space-between;

    margin: 20px;

}

.filters {

    width: 28%;

    padding: 10px;

    border: 1px solid #ccc;

}

.filter {

    margin-bottom: 10px;

    margin-right: 20px;

}

.car-list {

    width: 71%;

    display: flex;

    flex-wrap: wrap;

}

.car-card {

    width: 100%;

    border: 1px solid #ccc;

    margin: 10px;

    padding: 10px;

    display: flex;

}

.car-image {

    max-width: 420px;

    height: auto;

}

.car-description {

    margin-top: 10px;

}

.car-price {

    font-weight: bold;

    margin-top: 10px;

}

.car-info {

    margin-left: 5px;

}

js:

const carsData = [

    {

        brand: "Audi",

        model: "A4",

        year: 2018,

        price: 25000,

        image: "images/audi\_a4\_\_0009.webp",

        engine: "2.0 л, бензин",

        mileage: 40000,

        city: "Київ",

        transmission: "автомат",

    },

    {

        brand: "BMW",

        model: "3 Series",

        year: 2019,

        price: 28000,

        image: "images/bmw\_3s\_\_0762.webp",

        engine: "2.0 л, бензин",

        mileage: 35000,

        city: "Львів",

        transmission: "автомат",

    },

    {

        brand: "Mercedes-Benz",

        model: "C-Class",

        year: 2020,

        price: 32000,

        image: "images/mercedes-benz\_c-class\_\_522484991f.webp",

        engine: "2.0 л, дизель",

        mileage: 25000,

        city: "Одеса",

        transmission: "автомат",

    },

    {

        brand: "Toyota",

        model: "Camry",

        year: 2019,

        price: 23000,

        image: "images/toyota\_camry\_\_517156419f.webp",

        engine: "2.5 л, гібрид",

        mileage: 30000,

        city: "Дніпро",

        transmission: "автомат",

    },

    {

        brand: "Honda",

        model: "Civic",

        year: 2020,

        price: 21000,

        image: "images/honda\_civic\_\_521879904f.webp",

        engine: "1.5 л, бензин",

        mileage: 28000,

        city: "Харків",

        transmission: "автомат",

    },

    {

        brand: "Ford",

        model: "Focus",

        year: 2019,

        price: 19000,

        image: "images/ford\_focus\_\_522520559f.webp",

        engine: "1.0 л, бензин",

        mileage: 32000,

        city: "Кривий Ріг",

        transmission: "механіка",

    },

    {

        brand: "Volkswagen",

        model: "Golf",

        year: 2017,

        price: 17000,

        image: "images/volkswagen\_golf\_\_522443846f.webp",

        engine: "1.4 л, бензин",

        mileage: 42000,

        city: "Запоріжжя",

        transmission: "автомат",

    },

    {

        brand: "Nissan",

        model: "Altima",

        year: 2018,

        price: 16000,

        image: "images/nissan\_altima\_\_522418914f.webp",

        engine: "2.5 л, бензин",

        mileage: 38000,

        city: "Полтава",

        transmission: "автомат",

    },

    {

        brand: "Audi",

        model: "Q5",

        year: 2021,

        price: 35000,

        image: "images/audi\_q5\_\_522091412f.webp",

        engine: "2.0 л, бензин",

        mileage: 18000,

        city: "Львів",

        transmission: "автомат",

    },

    {

        brand: "Audi",

        model: "A3",

        year: 2020,

        price: 28000,

        image: "images/audi\_a3\_\_521530646f.webp",

        engine: "1.8 л, бензин",

        mileage: 22000,

        city: "Київ",

        transmission: "автомат",

    },

    {

        brand: "BMW",

        model: "5 Series",

        year: 2019,

        price: 32000,

        image: "images/bmw\_5-series\_\_426173066f.webp",

        engine: "3.0 л, бензин",

        mileage: 30000,

        city: "Одеса",

        transmission: "автомат",

    },

    {

        brand: "BMW",

        model: "X3",

        year: 2020,

        price: 35000,

        image: "images/bmw\_x3\_\_513474983f.webp",

        engine: "2.0 л, дизель",

        mileage: 25000,

        city: "Дніпро",

        transmission: "автомат",

    },

    {

        brand: "Mercedes-Benz",

        model: "E-Class",

        year: 2019,

        price: 33000,

        image: "images/mercedes-benz\_e-class\_\_522083588f.webp",

        engine: "2.0 л, бензин",

        mileage: 28000,

        city: "Харків",

        transmission: "автомат",

    },

    {

        brand: "Mercedes-Benz",

        model: "GLC",

        year: 2020,

        price: 36000,

        image: "images/mercedes-benz\_glc-class\_\_515876884f.webp",

        engine: "2.5 л, гібрид",

        mileage: 20000,

        city: "Львів",

        transmission: "автомат",

    },

    {

        brand: "Toyota",

        model: "RAV4",

        year: 2020,

        price: 28000,

        image: "images/toyota\_rav4\_\_521706951f.webp",

        engine: "2.5 л, бензин",

        mileage: 28000,

        city: "Київ",

        transmission: "автомат",

    },

    {

        brand: "Toyota",

        model: "Corolla",

        year: 2021,

        price: 30000,

        image: "images/toyota\_corolla\_\_522293872f.webp",

        engine: "1.8 л, гібрид",

        mileage: 22000,

        city: "Львів",

        transmission: "автомат",

    },

    {

        brand: "Honda",

        model: "Accord",

        year: 2020,

        price: 26000,

        image: "images/honda\_accord\_\_520697326f.webp",

        engine: "2.0 л, бензин",

        mileage: 25000,

        city: "Одеса",

        transmission: "автомат",

    },

];

const carList = document.querySelector(".car-list");

// Отримати унікальні марки та роки випуску з масиву автомобілів

const uniqueBrands = [...new Set(carsData.map((car) => car.brand))];

const uniqueYears = [...new Set(carsData.map((car) => car.year))];

// Отримати посилання на вибіркові списки в HTML

const brandFilter = document.getElementById("brandFilter");

const yearFilter = document.getElementById("yearFilter");

// Заповнити вибіркові списки унікальними марками та роками випуску

uniqueBrands.forEach((brand) => {

    const option = document.createElement("option");

    option.value = brand;

    option.textContent = brand;

    brandFilter.appendChild(option);

});

uniqueYears.forEach((year) => {

    const option = document.createElement("option");

    option.value = year;

    option.textContent = year;

    yearFilter.appendChild(option);

});

const priceFilterMin = document.getElementById("priceFilterMin");

const priceFilterMax = document.getElementById("priceFilterMax");

const mileageFilterMin = document.getElementById("mileageFilterMin");

const mileageFilterMax = document.getElementById("mileageFilterMax");

const transmissionFilter = document.getElementById("transmissionFilter");

const applyFiltersButton = document.getElementById("applyFilters");

function populateCarList() {

    // Отримати вибрані значення фільтрів

    const selectedBrand = brandFilter.value;

    const selectedYear = yearFilter.value;

    carList.innerHTML = "";

    carsData.forEach((car) => {

        // Додавання умов для фільтрації за обраними маркою і роком

        if (

            (selectedBrand === "all" || car.brand === selectedBrand) &&

            (selectedYear === "all" || car.year == selectedYear)

        ) {

            const carCard = document.createElement("div");

            carCard.classList.add("car-card");

            carCard.innerHTML = `

                <img src="${car.image}" class="car-image" alt="${car.brand}">

                <div class="car-info">

                    <h3>${car.brand} ${car.model}</h3>

                    <p>Ціна: $${car.price}</p>

                    <p>Рік: ${car.year}</p>

                    <p>Двигун: ${car.engine}</p>

                    <p>Пробіг: ${car.mileage} км</p>

                    <p>Місто: ${car.city}</p>

                    <p>Тип коробки передач: ${car.transmission}</p>

                </div>

            `;

            carList.appendChild(carCard);

        }

    });

}

function applyFilters() {

    const selectedBrand = brandFilter.value;

    const selectedYear = yearFilter.value;

    const selectedPriceMin = priceFilterMin.value;

    const selectedPriceMax = priceFilterMax.value;

    const selectedMileageMin = mileageFilterMin.value;

    const selectedMileageMax = mileageFilterMax.value;

    const selectedTransmission = transmissionFilter.value;

    const filteredCars = carsData.filter((car) => {

        return (

            (selectedBrand === "all" || car.brand === selectedBrand) &&

            (selectedYear === "all" || car.year == selectedYear) &&

            (selectedPriceMin === "" || car.price >= selectedPriceMin) &&

            (selectedPriceMax === "" || car.price <= selectedPriceMax) &&

            (selectedMileageMin === "" || car.mileage >= selectedMileageMin) &&

            (selectedMileageMax === "" || car.mileage <= selectedMileageMax) &&

            (selectedTransmission === "all" || car.transmission === selectedTransmission)

        );

    });

    carList.innerHTML = "";

    filteredCars.forEach((car) => {

        const carCard = document.createElement("div");

        carCard.classList.add("car-card");

        carCard.innerHTML = `

            <img src="${car.image}" class="car-image" alt="${car.brand}">

            <div class="car-info">

                <h3>${car.brand} ${car.model}</h3>

                <p>Ціна: $${car.price}</p>

                <p>Рік: ${car.year}</p>

                <p>Двигун: ${car.engine}</p>

                <p>Пробіг: ${car.mileage} км</p>

                <p>Місто: ${car.city}</p>

                <p>Тип коробки передач: ${car.transmission}</p>

            </div>

        `;

        carList.appendChild(carCard);

    });

}

applyFiltersButton.addEventListener("click", applyFilters);

populateCarList();