

ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ
Навчально-науковий інститут денної освіти
Форма навчання денна
Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту
Завідувач кафедри
_____ Олена ОЛЬХОВСЬКА
(підпис)

«___» _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему
**ІНФОРМАЦІЙНА СИСТЕМА З ПОШУКУ ТА ПІДБОРУ ТРАНСПОРТНОГО
ЗАСОБУ**

зі спеціальності 122 Комп'ютерні науки
освітня програма «Комп'ютерні науки»
ступеня магістра

Виконавець роботи Ломака Максим Анатолійович
_____ «___» _____ 2024 р.
(підпис)

Науковий керівник к.ф.-м.н. Ольховська Олена Володимирівна
_____ «___» _____ 2024 р.
(підпис)

Рецензент

ПОЛТАВА 2024 р.

ЗАТВЕРДЖУЮ
Завідувач кафедри _____ Олена ОЛЬХОВСЬКА
(підпис)
«___» _____ 2023 р.

ЗАВДАННЯ І КАЛЕНДАРНИЙ ГРАФІК ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

на тему «Інформаційна система з пошуку та підбору транспортного засобу»
зі спеціальності 122 Комп'ютерні науки
освітня програма «Комп'ютерні науки»
ступеня магістр

Прізвище, ім'я, по батькові Ломака Максим Анатолійович

Затверджена наказом ректора № ____-Н від «___» _____ 2023 р.

Термін подання студентом роботи «___» _____ 2024 р.

Вихідні дані до кваліфікаційної роботи: публікації з теми, навчальні тренажери в дистанційних курсах із комп'ютерних наук.

Зміст пояснювальної записки (перелік питань, які потрібно розробити)

ВСТУП

1. Постановка завдання
2. Інформаційний огляд
 - 2.1 Призначення та область застосування інформаційних систем з пошуку та підбору транспортного засобу
 - 2.2 Актуальність теми дослідження
 - 2.3 Порівняльний аналіз аналогів інформаційних систем з пошуку транспортного засобу
 - 2.4 Аналіз сайту «auto.ria.com»
 - 2.5 Аналіз сайту «rst.ua»
 - 2.6 Аналіз сайту «uavto»
 - 2.7 Аналіз сайту «ab.ua»
 - 2.8 Аналіз сайту «autoportal.ua»
 - 2.9 Аналіз порівнянь існуючих рішень розглянутих реалізацій
 - 2.10 Визначення вимог до інформаційної системи з пошуку та підбору транспортного засобу
3. Теоретична частина
 - 3.1 Обґрунтування вибору сучасних технологій веб-програмування для створення інформаційної системи з пошуку та підбору транспортного засобу
 - 3.2 Проектування структури бази даних
 - 3.3 Опис процесу проектування системи з пошуку та підбору транспортного засобу
 - 3.3.1 Опис предметної області
 - 3.3.2 Формування вимог до системи
 - 3.3.3 Проектування інтерфейсу користувача
 - 3.3.4 Проектування інформаційних систем з пошуку та підбору транспортного засобу
 - 3.4 Розробка структури системи «AUTOCHOOSE»
 - 3.5 Тестування та дослідницька експлуатація

3.5.1 Особливості тестування веб-орієнтованого програмного забезпечення

3.5.2 Перевірка сайту на кросбраузерність

3.5.3 Функціональне тестування програмного коду системи «AUTOCHOOSE»

4. Практична частина

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК А

Перелік графічного матеріалу: рисунки а також таблиці.

Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Постановка задачі	Ольховська О.В.		
Інформаційний огляд	Ольховська О.В.		
Теоретична частина	Ольховська О.В.		
Практична частина	Ольховська О.В.		

Календарний графік виконання кваліфікаційної роботи

Зміст роботи	Термін виконання	Фактичне виконання
Вступ		
Вивчення методичних рекомендацій та стандартів та звіт керівнику		
Постановка задачі		
Інформаційний огляд джерел бібліотек та інтернету		
Теоретична частина		
Практична частина		
Закінчення оформлення		
Доповідь студента на кафедрі		
Доробка (за необхідністю), рецензування		

Дата видачі завдання «__» _____ 2023 р.

Здобувач вищої освіти _____ Ломака Максим Анатолійович
(підпис)

Науковий керівник _____ к.ф.-н.м. Ольховська О.В.
(підпис) (науковий ступінь, вчене звання, ініціали та прізвище)

Результати захисту кваліфікаційної роботи

Кваліфікаційна робота оцінена на _____
(балів, оцінка за національною шкалою, оцінка за ECTS)

Протокол засідання ЕК № _____ від «__» _____ 2024 р.

Секретар ЕК _____
(підпис) (ініціали та прізвище)

Затверджую

Зав. кафедрою _____
к.ф.-м.н. Олена ОЛЬХОВСЬКА
« ____ » _____ 2023 р.

Погоджено

Науковий керівник _____
к.ф.-м.н. Олена ОЛЬХОВСЬКА
« ____ » _____ 2023 р.

План

дипломного проекту з фаху
спеціальності 122 Комп'ютерні науки
освітня програма 122 Комп'ютерні науки
на тему «Інформаційна система з пошуку та підбору транспортного засобу»
Прізвище, ім'я, по батькові Ломака Максим Анатолійович

ВСТУП

1. Постановка завдання
2. Інформаційний огляд
 - 2.1 Призначення та область застосування інформаційних систем з пошуку та підбору транспортного засобу
 - 2.2 Актуальність теми дослідження
 - 2.3 Порівняльний аналіз аналогів інформаційних систем з пошуку транспортного засобу
 - 2.4 Аналіз сайту «auto.ria.com»
 - 2.5 Аналіз сайту «rst.ua»
 - 2.6 Аналіз сайту «uavto»
 - 2.7 Аналіз сайту «ab.ua»
 - 2.8 Аналіз сайту «autoportal.ua»
 - 2.9 Аналіз порівнянь існуючих рішень розглянутих реалізацій
 - 2.10 Визначення вимог до інформаційної системи з пошуку та підбору транспортного засобу
3. Теоретична частина
 - 3.1 Обґрунтування вибору сучасних технологій веб-програмування для створення інформаційної системи з пошуку та підбору транспортного засобу
 - 3.2 Проектування структури бази даних
 - 3.3 Опис процесу проектування системи з пошуку та підбору транспортного засобу
 - 3.3.1 Опис предметної області
 - 3.3.2 Формування вимог до системи
 - 3.3.3 Проектування інтерфейсу користувача
 - 3.3.4 Проектування інформаційних систем з пошуку та підбору транспортного засобу
 - 3.4 Розробка структури системи «AUTOCHOOSE»
 - 3.5 Тестування та дослідницька експлуатація
 - 3.5.1 Особливості тестування веб-орієнтованого програмного забезпечення
 - 3.5.2 Перевірка сайту на кросбраузерність
 - 3.5.4 Функціональне тестування програмного коду системи «AUTOCHOOSE»

4. Практична частина

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК А

Здобувач вищої освіти _____ Ломака М. А.

(підпис)

«_____» _____ 2023 р.

РЕФЕРАТ

Записка: 72 с., 24 рис., 11 таблиць, 13 використаних джерел

Актуальність теми. Тема дослідження є досить актуальною, оскільки щоденно в різних регіонах країни зростає попит на придбання транспортних засобів. Це обумовлено покращенням процесу пошуку та підбору транспортного засобу, що не призводить до втрати часу та довіри адміністраторів порталів, а представить зручний вибір пропозицій під бажані потреби користувачів.

Мета роботи і задачі дослідження: Мета кваліфікаційної роботи – підвищення ефективності, зручності й надійності з пошуку та підбору транспортного засобу за заданими критеріями, шляхом розробки веб-сайту, призначеного для збирання та аналізу даних з різних популярних порталів придбання авто.

Для досягнення поставленої мети в роботі розв’язуються наступні завдання:

- провести порівняльний аналіз існуючих порталів пошуку транспортного засобу;
- здійснити вибір програмно-технічних засобів для реалізації завдань дослідження;
- спроектувати систему з пошуку та підбору транспортного засобу у вигляді сайту «AUTOCHOOSE»;
- розробити базу даних сайту «AUTOCHOOSE»;
- розробити сайт з пошуку та підбору транспортного засобу «AUTOCHOOSE»;

Об’єкт дослідження: процес прийняття рішення щодо пошуку та підбору транспортного засобу.

Предмет дослідження: інформаційна система з пошуку та підбору транспортного засобу.

Методи дослідження. В роботі застосовувалися такі методи: теоретичні (аналіз літератури з проблеми дослідження та вивчення технологій програмування і програмних засобів, необхідних для створення інформаційної системи з пошуку та підбору транспортного засобу); спеціальні методи (аналіз предметної області, проектування і реалізація сайту «AUTOCHOOSE»);

Перелік ключових слів: ІНФОРМАЦІЙНА СИСТЕМА, ПОШУК, ПІДБІР, ТРАНСПОРТНИЙ ЗАСІБ, ПРОГРАМНІ ЗАСОБИ.

ЗМІСТ

ВСТУП.....	9
1. Постановка завдання	10
2. Інформаційний огляд та порівняльний аналіз аналогів інформаційних систем з пошуку та підбору транспортного засобу	11
2.1 Призначення та область застосування інформаційних систем з пошуку та підбору транспортного засобу.....	11
2.2 Актуальність теми дослідження	12
2.3 Порівняльний аналіз аналогів інформаційних систем з пошуку транспортного засобу	13
2.4 Аналіз сайту «auto.ria.com»..	15
2.5 Аналіз сайту «rst.ua».	16
2.6 Аналіз сайту «uavto».....	17
2.7 Аналіз сайту «ab.ua».	17
2.8 Аналіз сайту «autoportal.ua».	19
2.9 Аналіз порівнянь існуючих рішень розглянутих реалізацій.	19
2.10 Визначення вимог до інформаційної системи з пошуку та підбору транспортного засобу.....	21
3. Теоритична частина	22
3.1 Обґрунтування вибору сучасних технологій веб-програмування для створення інформаційної системи з пошуку та підбору транспортного засобу.	22
3.2 Проектування структури бази даних.	35
3.3 Опис процесу проектування системи з пошуку та підбору транспортного засобу. .41	
3.3.1 Опис предметної області.....	42
3.3.3 Проектування інтерфейсу користувача.	44
3.3.4 Проектування інформаційних систем з пошуку та підбору транспортного засобу.	45
3.4 Розробка структури системи «AUTOCHOOSE».....	48

3.5 Тестування та дослідницька експлуатація	49
3.5.1 Особливості тестування веб-орієнтованого програмного забезпечення.....	49
3.5.2 Перевірка сайту на кросбраузерність	51
3.5.3 Функціональне тестування програмного коду системи «AUTOCHOOSE» ...	52
4. Практична частина.....	56
ВИСНОВКИ.....	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	62
ДОДАТОК А	63

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ**

1. БД - база даних;
2. СУБД - система управління базами даних;
3. CMS - Content Management System;
4. ПЗ - програмне забезпечення;
5. HTML - HyperText Markup Language;
6. PHP - скриптова мова загального призначення;
7. CSS - Cascading Style Sheets.

ВСТУП

За останні роки відбулись значні зміни в сфері інформаційної діяльності та послуг. На зміну традиційним підходам підбору транспортного засобу з подання оголошень автолюбителями та агентствами в газетах появився віртуальний та інтерактивний перегляд пошуку на інтернет-порталах. Безумовно, що зараз існує багато ресурсів пошуку та підбору транспортного засобу.

Актуальність роботи. Тема дослідження є досить актуальною, оскільки щоденно в різних регіонах країни зростає попит на придбання транспортних засобів. Це обумовлено покращенням процесу пошуку та підбору транспортного засобу, що не призводить до втрати часу та довіри адміністраторів порталів, а представляє зручний вибір пропозицій під бажані потреби користувачів.

Мета кваліфікаційної роботи - підвищення ефективності, зручності й надійності пошуку та підбору транспортного засобу за заданими критеріями, шляхом розробки веб-сайту, призначеного для збирання та аналізу даних з різних популярних порталів придбання авто.

Для досягнення поставленої мети в роботі розв'язуються наступні завдання:

- провести порівняльний аналіз існуючих порталів пошуку транспортного засобу;
- здійснити вибір програмно-технічних засобів для реалізації завдань дослідження;
- спроектувати інформаційну систему з пошуку та підбору транспортного засобу у вигляді сайту «AUTOCHOOSE»;
- розробити базу даних сайту «AUTOCHOOSE»;
- розробити сайт з пошуку та підбору транспортного засобу «AUTOCHOOSE»;

Об'єктом дослідження є процес прийняття рішення щодо пошуку та підбору транспортного засобу.

Предметом дослідження є інформаційна система з пошуку та підбору транспортного засобу.

1. Постановка завдання

Метою та завданням кваліфікаційної роботи є підвищення ефективності, зручності й надійності з пошуку та підбору транспортного засобу за заданими критеріями, шляхом розробки веб-сайту за темою «Інформаційна система з пошуку та підбору транспортного засобу».

Робота цього веб-сайту повинна бути реалізована через систему пошуку в ньому.

2. Інформаційний огляд та порівняльний аналіз аналогів інформаційних систем з пошуку та підбору транспортного засобу

2.1 Призначення та область застосування інформаційних систем з пошуку та підбору транспортного засобу.

На сьогодні відбуваються радикальні зміни в соціальній структурі світу, які торкнулись також і розширення сфери інформаційної діяльності та послуг. У такій сфері працює приблизно половина від загальної кількості зайнятого населення. До цієї сфери все більше стає наближеною і система послуг з пошуку та придбання авто. Однією з найскладніших задач, з якою доводиться стикатися при організації роботи таких компаній, є управління послугами, які вона надає.

Аналіз сфери послуг аналогічної спрямованості показав, що особливо актуальні для різних сфер економічної діяльності дослідження, які проводяться за допомогою сучасних інформаційних технологій. Галузі впровадження інформаційних технологій пов'язані зі створенням продукції і наданням послуг, з вивченням ринків збуту, конкурентів, постачальників, покупців і т.д. При пошуку можливостей для впровадження інформаційних технологій на першому місці стоять не технічні питання, а питання підвищення ефективності роботи компанії і підвищення її виробничого потенціалу за допомогою інформаційних технологій. Продукції та послуги – це перша сфера застосування інформаційних технологій.

Для розвитку підприємств з продажу автомобілів в гострій конкурентній боротьбі, необхідно володіти якісною інформацією: знати стан сучасного авторинку, тенденції обслуговування автомобілів, брати найкраще з досвіду колег, зарубіжних партнерів. Тоді процес бізнес-планування компанії буде виконуватись без зайвих витрат та з високим рівнем продуктивності.

Сьогодні автомобільний ринок України переживає не найкращі свої часи. При цьому українці частіше купують саме вживані автомобілі, а не нові.

Людина живе в активному ритмі, тому далеко не завжди є час, щоб вивчати газети та журнали з оголошеннями, ходити по різних салонах і так далі.

Безсумнівно, що нині існує багато способів з пошуку та придбання

транспортного засобу. Але й бажання користувачів різноманітні. Кожен з них ставить перед собою мету щодо придбання тієї чи іншої марки транспортного засобу, керуючись бажаними критеріями вибору та реальними фінансовими можливостями.

Добрими помічниками при цьому стають інформаційно-консультаційні системи з пошуку та підбору транспортного засобу. Це спеціальні веб-ресурси, через які можна купити та продати транспортний засіб. Помилково вважати, що сайти продажу орієнтовані лише на вживані авто. На великих ресурсах автосалони заводять власні сторінки, регулярно оновлюють список доступних транспортних засобів, що спрощує вибір потенційного покупця.

Також це чудова можливість донести до максимально великої кількості людей свій намір продати транспортний засіб. Публікуючи оголошення онлайн на популярному сайті з пошуку та підбору транспортного засобу, кількість переглядів значно перевищує охоплення місцевої газети, тематичного журналу або наклеєного на заднє скло самої машини оголошення.

Звісно, пошук транспортного засобу сьогодні найпопулярніший у великих містах. Тут на сервісах з пошуку та підбору транспортного засобу можна згідно фільтру переглянути всі запропоновані моделі, їх вартість, новизну, характеристики та інше. Користувач за допомогою фільтру та пошуку може знайти відповідні позиції, вибрати їх, заповнити форму з особистими даними та відправити для обробки.

На даний час в невеличкому місті чи регіоні чимало існує компаній, які хочуть мати інформаційно-консультаційну систему у вигляді власного ресурсу, платформу чи майданчик в зручному графічному представленні, розміщеним у мережах, де задаються всі необхідні параметри за результатами пошуку та підбору транспортного засобу.

2.2 Актуальність теми дослідження

Актуальність теми дослідження, що розглядається в кваліфікаційній роботі магістра, обумовлена необхідністю створення зручного і прийняттого для користувача інтерфейсу системи у вигляді сайту, який надасть можливість

людям, які не володіють навичками роботи у мережі Інтернет пришвидшити пошук та підбір необхідного транспортного засобу.

Для досягнення даної мети необхідно провести огляд програмних засобів для розробки даної системи та аналіз переваг й недоліків вже існуючих систем з пошуку та підбору транспортного засобу, вибрати мову програмування та провести тестування. Все повинно бути реалізоване в зручному графічному представленні, що може бути як локальним, так і розміщеним у мережах, де задаються всі необхідні параметри за результатами пошуку й підбору транспортного засобу з вибраними найкращими результатами, таблицею з розбірливою прогнозованою вартістю та основними характеристиками транспортного засобу.

2.3 Порівняльний аналіз аналогів інформаційних систем з пошуку транспортного засобу

...Перш, ніж створити власну інформаційну систему у вигляді веб-сайту, потрібно знати переваги та недоліки вже існуючих рішень з пошуку транспортного засобу.

Серед основних переваг інформаційних ресурсів з пошуку транспортного засобу можна виділити саме доступність такого пошуку, адже більшість з них є безкоштовними. Але крім переваг, їм властиві й певні недоліки.

Головним їх недоліком в більшості є представлення не зовсім зручного інтерфейсу; опису всіх характеристик та критеріїв вибору, притаманних майже кожному користувачу; відсутність компетентного консультанта щодо підбору бажаного транспортного засобу. Тому, для пошуку транспортного засобу в мережах, згідно бажаних критеріїв та реальних можливостей користувача, потребується чимало часу.

Дослідивши велику кількість інформаційних систем з пошуку та підбору необхідного транспортного засобу у вигляді сайтів, визначено найкращі варіанти та виокремлено особливості кожної з них для проектування власної розробки:

- сайт «auto.ria.com»;
- сайт «rst.ua»;
- сайт «uavto.com.ua»;

- сайт «ab.ua»;
- сайт «autoportal.ua».

2.4 Аналіз сайту «auto.ria.com». В ході виконання курсової роботи першим проаналізовано сайт «auto.ria.com». Сайт з пошуку та підбору необхідного транспортного засобу «auto.ria.com» знаходиться за адресою: <https://auto.ria.com/uk/>.

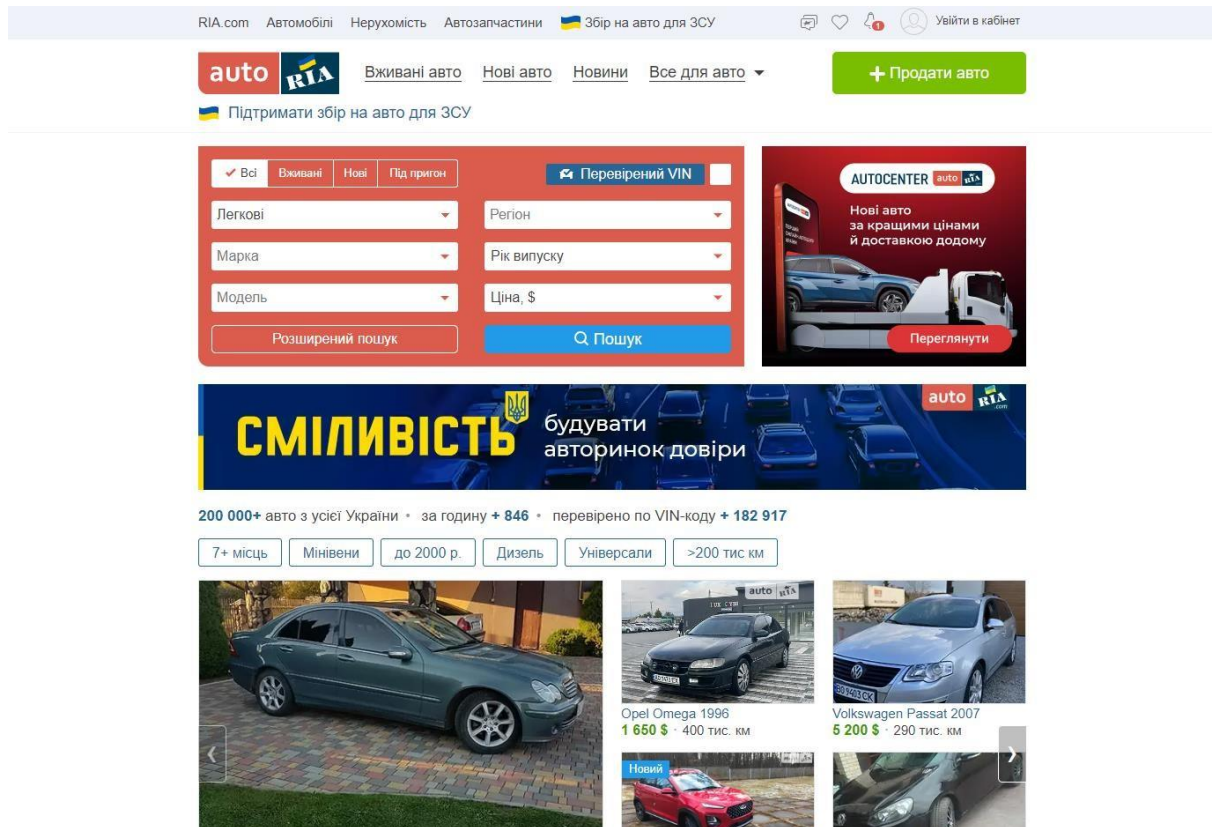


Рисунок 2.4.1 – Головна сторінка сайту «auto.ria.com»

Об'єктивно, що сайт «auto.ria.com» це найкращий ресурс. Практично вся автомобільна Україна, коли потрібно продати авто чи знайти його, обираючи найкращі сайти, пальму першості віддає саме цьому ресурсу. У нього є відгалуження, де здійснюється продаж та купівля.

Основний акцент сайту «auto.ria.com» зроблено я на автомобільному сегменті проекту. Для роботи із сайтом потрібно обов'язково пройти реєстрацію, після чого виконати авторизацію. Крім вживаних автомобілів, тут можна знайти ломбарди, розбирання, автовикупи, запчастини та автосервіси.

«auto.ria.com» – це агрегатор оголошень з продажу автомобілів. Оголошення можуть бути платними та безкоштовними. Сайту притаманні відмінна технічна підтримка та добре продумане меню. Працювати з таким сайтом досить приємно, але іноді спостерігається сильна завантаженість, через

що ресурс відверто гальмує. Не дивлячись на певні недоліки сайту, він вважається одним з найкращих автомобільних сайтів із продажу автомобілів в Україні 2021 року.

2.5 Аналіз сайту «rst.ua». Другим в роботі проаналізовано сайт «rst.ua».

Веб-сторінка сайту «rst.ua» знаходиться за адресою: <https://rst.ua/ukr/>.

Головна сторінка сайту «rst.ua» зображена на рисунку 2.5.1.

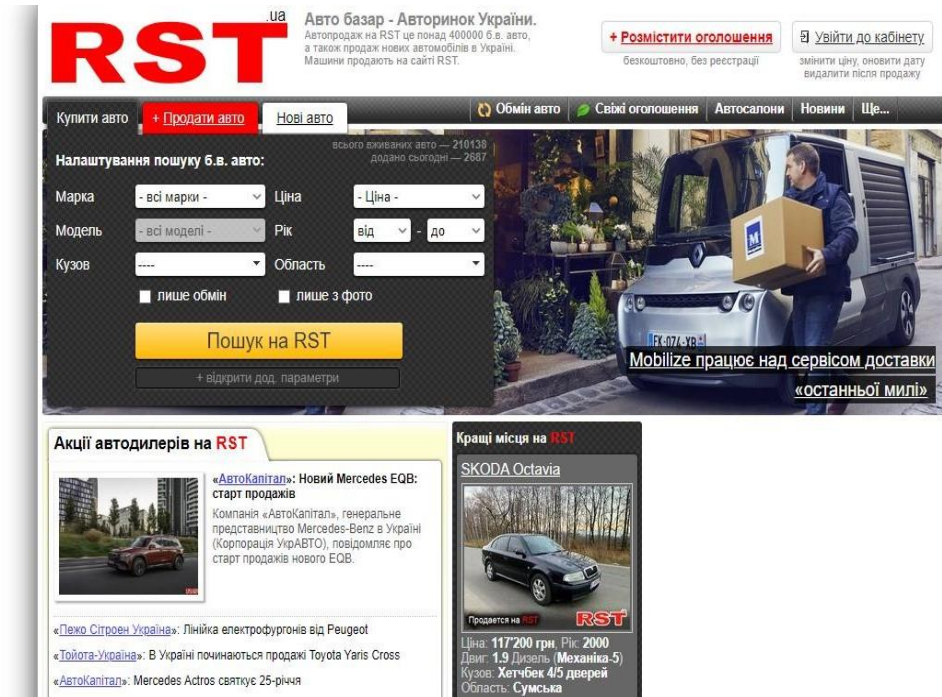


Рисунок 2.5.1 – Головна сторінка сайту «rst.ua»

Другим по популярності є сайт «rst.ua». Хоча багато хто вважає, що RST краще Авто Ріа за рядом параметрів. Але, якщо спиратися на популярність та кількість користувачів й актуальних оголошень, то друга позиція буде справедливою та закономірною. Тут все працює дещо інакше. Доступний широкий каталог машин, що продаються, списки авторинків країни, розділ з новинами, новинками автомобільного світу і так далі. Опублікувати оголошення можна без реєстрації. Продавати авто дозволяється за допомогою безкоштовних оголошень.

Варто відмітити, що дуже високо користувачі оцінили систему фільтрації пошуку сайту, де є близько 20 різних параметрів і критеріїв. Служба підтримки працює чудово, є телефонні помічники, а також онлайн консультанти.

На мою думку, з цього сайту є дещо, що можна використати для власної розробки інформаційної системи з пошуку та підбору транспортного засобу.

2.6 Аналіз сайту «uavto». Розглянемо для порівняння сайт «uavto». Він знаходиться за адресою: <http://uavto.com.ua>. Головна сторінка сайту «uavto» зображена на рисунку 2.6.1.



Рисунок 2.6.1 – Головна сторінка сайту «uavto»

Сайт «uavto.com.ua» є досить популярним. Він пропонує близько 250 тисяч оголошень та дозволяє шукати автомобілі регіонами. Сайт здебільшого спеціалізується на автомобілях та мототехніці. На головній сторінці сайту (рис. 2.3.1) є карта та список регіонів, що відразу ж спрощує пошук потрібних варіантів.

Тут пропонують продаж нових та вживаних автомобілів, вантажівок, водний транспорт, сільськогосподарську техніку. Сайт «uavto.com.ua» виглядає зовсім просто, але забезпечує комфортну навігацію. Приємним враженням є те, що навіть не заважає нав'язлива реклама – проста, але зі смаком.

2.7 Аналіз сайту «ab.ua». Розглянемо також для порівняння сайт «ab.ua». Він знаходиться за адресою: <https://ab.ua>. Головна сторінка сайту «ab.ua» зображена на рисунку 2.7.1.

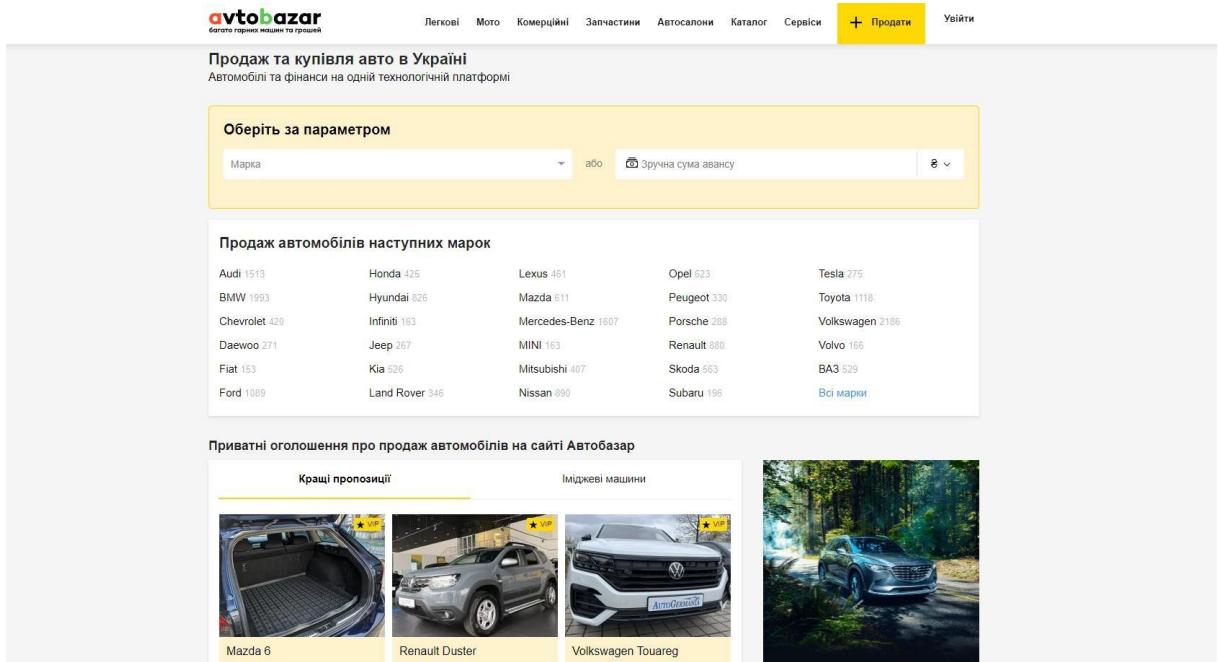


Рисунок 2.7.1 – Головна сторінка сайту «ab.ua»

Ще кілька років тому сайт «ab.ua» займав лідируючі позиції серед інформаційних ресурсів з пошуку та підбору транспортного засобу, але поступово почав втрачати популярність.

Хоча говорити про повне зникнення ресурсу точно не доводиться. Щодня там з'являється близько 2000 нових оголошень. Варто зазначити, що представництва автосалонів та магазини запчастин тут відсутні. Проте, на ньому можна знайти різні види транспорту, включаючи авіацію. Щодо інтерфейсу, то не зовсім він є привабливим для користувача. Більше всього, для власної розробки почерпнути з нього немає чого.

2.8 Аналіз сайту «autoportal.ua». Розглянемо для порівняння сайт «autoportal.ua». Він знаходиться за адресою: <https://autoportal.ua>. Головна сторінка сайту «autoportal.ua» зображена на рисунку 2.8.1.

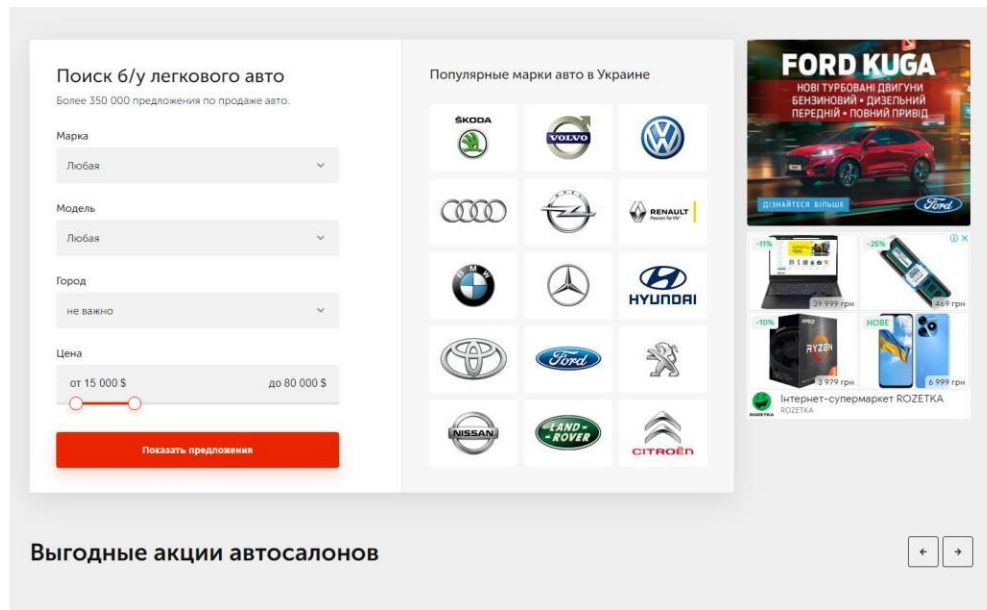


Рисунок 2.8.1 – Головна сторінка сайту «autoportal.ua»

«autoportal.ua» має непогані перспективи сьогодення на українському авторинку – цікавий, зручний у плані навігації сайт з пошуку та підбору транспортного засобу. Перевагою його є те, що він за кількістю оголошень поступається популярнішим ресурсам.

Сайту притаманна зручна система фільтрації за кількістю параметрів, а також є окремий розділ з інформацією про вживані автомобілі. Причому, щодня сюди додається понад 100 нових публікацій із цікавими пропозиціями. Як основний сайт для продажу, може і не варто використовувати, але для максимального охоплення аудиторії, закинути оголошення на «autoportal.ua» точно варто. На мою думку, доцільніше було б, щоб сайт був україномовним.

2.9 Аналіз порівнянь існуючих рішень розглянутих реалізацій. У таблиці 2.9.1 представлено порівняльний аналіз розглянутих сайтів, а саме: «auto.ria.com», «rst.ua», «uavto.com.ua», «ab.ua», «autoportal.ua».

У якості критеріїв для порівняння було обрано такі критерії: дизайн, швидкість завантаження, можливість задати питання онлайн, можливість відслідковувати вартість авто, наявність новин та відгуків про сайт.

Таблиця 2.9.1 – Таблиця порівнянь існуючих рішень розглянутих реалізацій

Назва сайту	Критерії порівняння сайтів				
	Дизайн (0-5)	Швидкість завантаження (0-5)	Можливість задати питання онлайн	Можливість відслідкову- вати вартість авто	Наявність новин та відгуків про сайт
«auto.ria.com»	5	4	+	+	+
«rst.ua»	2	3	+	—	—
«uavto.com.ua»	3	4	—	—	+
«ab.ua»	4	2	+	+	+
«autoportal.ua»	3	3	—	—	—

Підсумовуючи перегляд та оцінку приведених вище сайтів, варто зазначити, що всі вони мають наступні характеристики: це є сайт, який містить в собі стандартні функції з пошуку та підбору транспортного засобу. Кожен користувач має можливість швидко знайти бажаний транспортний засіб, який задовольняє його вимоги. Також існує можливість якісно та кількісно оцінити окремі моделі транспортного засобу. Варто зауважити, що інтерфейс деяких з них не завжди є інтуїтивно зрозумілим та доступним новачку, неукраїномовний, з непривабливою гамою кольорів.

Проаналізувавши існуючі рішення розглянутих реалізацій, в розроблювальній інформаційної системи у вигляді сайту «AUTOCHOOSE» пропонується звернути увагу на такі моменти:

- сайт «AUTOCHOOSE» повинен мати зручний, зрозумілий та приємний інтерфейс;
- сайт «AUTOCHOOSE» повинен мати швидкість завантаження серед оцінок (0-5), а саме: 4;
- на сайті «AUTOCHOOSE» передбачено можливість задати питання онлайн;
- на сайті «AUTOCHOOSE» передбачено можливість відслідковувати вартість транспортного засобу;

- сайт «AUTOCHOOSE» повинен мати наявність новин та відгуків про сайт.

2.10 Визначення вимог до інформаційної системи з пошуку та підбору транспортного засобу.

Основною метою розробки курсової роботи є концепція єдності, щоб вся необхідна для роботи інформація зберігалася на єдиному веб-сервісі.

Інформаційна система з пошуку та підбору транспортного засобу має являти собою веб-сайт «AUTOCHOOSE» і повинна бути кросбраузерною та не вимогливою до системних ресурсів. Інтерфейс системи має бути мінімалістичним, сучасним та інтуїтивно зрозумілим.

В таблиці 2.7.1 представлено перелік мінімальних версій браузерів, які має підтримувати розроблена система.

Таблиця 2.7.1 – Перелік мінімальних версій браузерів, яких має підтримувати система

Браузери	Мінімальні версії
Chrome	49+
Firefox	52+

В інформаційній системі з пошуку та підбору транспортного засобу «AUTOCHOOSE» потрібно реалізувати наступні функції:

- можливість реєстрації облікового запису;
- можливість входу до облікового запису;
- пошук оголошень за параметрами;
- додавання нових оголошень;
- детальний опис моделі транспортного засобу та його виробника;
- зворотний зв'язок;
- дані про технічний стан транспортного засобу;
- додавання новин порталу.

3. Теоритична частина

3.1 Обґрунтування вибору сучасних технологій веб-програмування для створення інформаційної системи з пошуку та підбору транспортного засобу.

Для створення інформаційної системи з пошуку та підбору транспортного засобу «AUTOCHOOSE», використано такі засоби:

- HTML, CSS;
- backend мова програмування: PHP;
- frontend мова програмування: JavaScript;
- асинхронна технологія AJAX;
- локальний сервер Open Server Panel;
- система управління базою даних phpMyAdmin;
- база даних: MySQL.

В якості backend мови програмування був обраний PHP версії 7.1.

PHP – скриптова мова загального призначення, що інтенсивно застосовується для розробки Web-додатків. На даний момент взаємодіє з великою кількістю хостинг-провайдерів та є одним із лідерів серед мов програмування, що використовуються для розробки Web-сайтів. Ця мова програмування створена безпосередньо для ведення Web-розробок і може впроваджуватись у програмний код Web-сторінки.

Основна відмінність цієї мови скриптів від скриптів, які написані, наприклад, на C++ або Perl – це те, що замість програми, що формує HTML-код, створюється програмний код із кількома впровадженими командами PHP. Для впровадження коду PHP його відокремлюють спеціальними початковим та кінцевим тегами, за допомогою яких процесор PHP визначає початок та кінець ділянки HTML-коду, який містить PHP-команди.

Мова програмування PHP містить у собі безліч інтегрованих функцій, таких як: функція роботи з файловою системою, функція роботи з протоколом HTTP, функція роботи з датою та часом, функція обробки рядків та масивів і т.д. Кожен програмний продукт, який написано на мові PHP, складається з набору спеціальних конструкцій. В якості таких конструкцій можуть служити будь-які

елементи, що використовуються в коді PHP, такі як цикли, оператори, функції та ін. Головна особливість цієї мови полягає в тому, що вона добре взаємодіє з усіма сучасними Web-технологіями та підтримує велику кількість сучасних Web-протоколів, а саме: протокол FTP, протокол POP, протокол SNMP і т.д. Також мова PHP чудово підходить для роботи з базами даних. У більшості сучасних СУБД здійснено підтримку скриптової мови програмування PHP.

Скрипт, написаний мовою PHP, може включати і 10 000 рядків коду і складатися з одного рядка – все полягає в тому, яке саме завдання необхідно вирішити з допомогою цього скрипта. Для початку виконання коду PHP його необхідно позначити спеціальним символом (тегом), який починатиме послідовність дій (<?) і продовжуватиме виконання цих дій до закриває тега (?>).

Для розробників PHP забезпечує гнучкі та ефективні засоби забезпечення безпеки, які умовно можна поділити на дві групи: засоби системного рівня та засоби рівня програми. Механізми безпеки, що реалізовані в мові PHP, знаходяться під керуванням адміністраторів, і можуть забезпечити максимальну безпеку при правильному налаштуванні. Також, у скриптовій мові PHP реалізовано так званий безпечний режим (safe mode), за допомогою якого обмежується можливість застосування коду певними категоріями користувачів. Наприклад, за допомогою безпечного режиму можливо лімітувати максимальний час, необхідний для виконання скрипту, а також обсяг використовуваної пам'яті, оскільки неконтрольована витрата пам'яті негативно позначається швидкодії Web-сервера. Крім цього, адміністратор може встановити обмеження використання каталогів, за допомогою яких користувач переглядає інформацію та виконує скрипти PHP, а також використовує скрипти PHP для перегляду конфіденційної інформації на веб-сервері.

До складу функцій мови PHP входить низка надійних механізмів шифрування. Також скрипти, написані мовою PHP, сумісні з багатьма програмами незалежних компаній. Це дозволяє легко інтегрувати його із захищеними технологіями електронної комерції.

Інша перевага цієї мови полягає в тому, що вихідний текст скрипту PHP не можна переглядати в браузері через те, що він компілюється безпосередньо перед відправкою на запит користувача. В результаті, таке виконання сценарію не допускає розкрадання тексту оригінальних кодів PHP команд користувачами. Стосовно потреб розробника мова PHP має гнучкість та ефективність налаштування.

Оскільки PHP не містить коду, який був би орієнтований на конкретний Web-сервер, то він чудово взаємодіє із сучасними серверами, а саме: сервер Microsoft IIS, сервер Netscape Enterprise Server, сервер Apache, сервер Stronghold та інші.

В якості frontend мови програмування для динамічного відображення інформації на сайті був обраний JavaScript.

JavaScript – це мова програмування, яка служить для керування сценаріями перегляду Web-сторінок. Головна особливість даної мови полягає в тому, що при її використанні є можливість змінювати властивості середовища відображення під час перегляду Web-сайту і при цьому не відбуватиметься перезавантаження Web-сторінок. Наприклад, за допомогою мови JavaScript можна замінити колір фону Web-сторінки або замінити зображення, інтегроване в Web-сторінку, також має можливість створити нове вікно відображення або вивести різні повідомлення.

JavaScript є об'єктно-орієнтованою мовою програмування, але в ній використовується прототипування (швидка реалізація базової функціональності для аналізу роботи системи), тому є ряд особливостей, пов'язаних із безпосереднім виконанням коду мови в порівнянні з традиційними об'єктно-орієнтованими мовами програмування. Крім цього, мова JavaScript має перелік властивостей, які властиві іншим функціональним мовам, а саме: функції як об'єкти першого класу, об'єкти як списки, анонімні функції.

Хоча мова JavaScript по синтаксису схожа на мову C, вона все ж має ряд відмінностей:

- використовуються об'єкти із здатністю самоаналізу;

- функції, як об'єкти першого класу;
- автоматичне наведення типів;
- анонімні функції.

В якості недоліків мови JavaScript можна виділити такі:

- немає можливості регулювати області;
- відсутність інтерфейсу;

використання стандартних інтерфейсів доступу до Web-серверів та баз даних.

Для покращення зовнішнього вигляду сайту була використана формальна мова опису CSS.

CSS або каскадні таблиці стилів – це формальна мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки. CSS використовується розробниками Web-сайтів для встановлення шрифтів, їх розташування на сторінці, кольорів, що використовуються в ході розробки та інших принципів зовнішнього вигляду блоків Web-сторінки.

Головним завданням, яке вирішується під час створення CSS є поділ вмісту, що, у свою чергу, дозволяє підвищити доступність документа, а також забезпечити більшу гнучкість та можливість управління поданням Web-сторінки та знизити складність структури її вмісту.

Крім цього, каскадні таблиці стилів дають можливість представити одну і ту ж Web-сторінку в різноманітних стилях відображення, наприклад, екранне уявлення, читання голосом тощо. CSS складається з набору правил. У свою чергу, кожне правило включає один або кілька перемикачів, які розділені комами, і блок визначень, що полягає у фігурних дужках ({}), і складається з набору властивостей та їх значень.

При безпосередньому відображенні Web-сторінки таблицю стилів можна взяти з різних джерел. Ієрархія стилів складається з:

- авторські стилі (інформація стилів, що надається розробником сторінки);
- стилі користувача;
- стилі браузера.

На додаток до цього, каскадні таблиці стилів надають можливість працювати зі шрифтовим оформленням документа на більш високому рівні, ніж стандартний HTML-код, уникаючи при цьому обтяження сторінок графікою.

В якості системи управління базами даних був обраний MySQL.

СУБД MySQL. На даний момент MySQL є однією з найнадійніших, швидких, якісних та відомих із усіх існуючих сучасних систем керування базами даних. Основною причиною цього є її безкоштовне розповсюдження разом зі своїми вихідними кодами, інша причина – те, що MySQL досить швидка СУБД.

Головна особливість роботи СУБД MySQL полягає у використанні мови структурованих запитів – SQL у ролі основного в управлінні базою даних, а саме: для створення або видалення таблиць у базі даних, здійснення вибірки з бази даних для безпосереднього заповнення таблиць даними.

MySQL має високу стійкість, високу швидкість роботи, простотою в налаштуванні та використанні, вихідники сервера компілюються на безлічі платформ, тому СУБД MySQL є гідним рішенням для невеликих програм.

Основні можливості MySQL:

- надає можливість одночасної роботи з базою даних необмежену кількість користувачів;
- кількість рядків у таблицях може сягати 50 млн;
- висока швидкість виконання команд користувачів;
- проста та ефективна система безпеки.

В якості ще однієї переваги перед іншими СУБД можна виділити те, що MySQL може працювати з мовою SQL у стандарті ANSI 92, а також має безліч встановлюваних розширень до цього стандарту, яких не передбачено в жодній іншій системі управління базами даних.

Для покращення роботи з клієнт-серверними запитами була використана технологія асинхронної передачі даних AJAX.

Технологія асинхронної передачі даних AJAX. Багато технологій, що використовуються в AJAX, відомі ще з 1990-х років. Так, 1996 року в Internet

Explorer 3 застосовувався HTML-елемент IFRAME, а 1998 року компанія Microsoft запропонувала підхід Remote Scripting.

Безпосередньо термін AJAX вперше був використаний Джессі Джеймсом Гарретом 18 лютого 2005 року у статті «Ajax: A New Approach to Web Applications». Її автор є одним із засновників та головою компанії Adaptive Path. У статті він описав принцип розробки web-додатків, застосовуваний на той час у Google Maps і Gmail. За його словами, це стало «фундаментальним проривом у можливостях, доступних у веб-додатках».

Тоді ж Гаррет пояснив, що таке AJAX, дав назву цьому підходу і звернув увагу безпосередньо на тренд, що виник. Все це дозволило вивести розробку Web-додатків на новий рівень. Тепер можна на власному дисплеї спостерігати результати фонових обмінів даними браузера з сервером.

За механізмом роботи сторінки відразу можна сказати, що це сайт AJAX. Раніше користувач повинен був натискати на кнопки і переходити на посилання для підтвердження своїх дій. А тепер сторінка сама реагує на внесення даних належним чином. В результаті час, витрачений на спілкування із сайтом, помітно скорочується. Користувач спілкується з веб-програмою, що швидко реагує. Для його нормальної роботи достатньо браузера, що підтримує JavaScript та підключення до Інтернету.

Варто зазначити, що правильно налаштований веб-додаток під час використання цієї технології функціонуватиме аналогічно звичайній програмі на комп'ютері.

Зрозуміти основний принцип роботи AJAX (рис. 3.1.1) допомагає наведене нижче зображення.

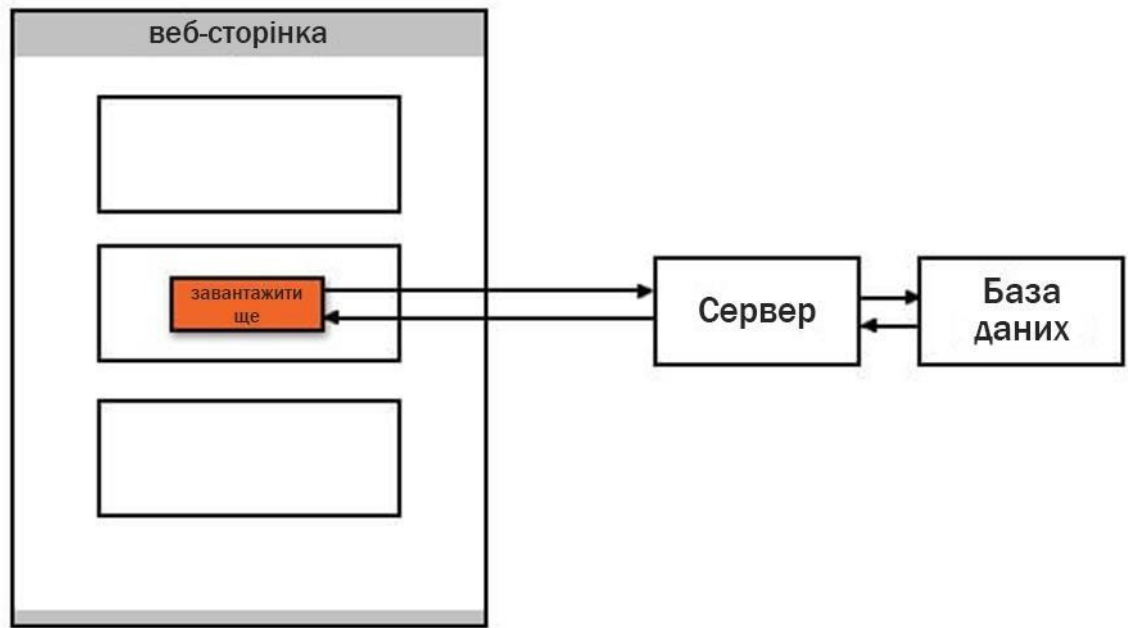


Рисунок 3.1.1 – Основний принцип роботи AJAX

У роботі технології можна виділити 4 основні етапи:

- 1) Користувач викликає AJAX. Зазвичай, це реалізується за допомогою будь-якої кнопки, яка пропонує отримати більше інформації.
- 2) Система надсилає на сервер запит та різноманітні дані. Наприклад, може знадобитися завантаження певного файлу чи конкретних відомостей із бази даних.
- 3) Сервер отримує відповідь від бази даних та надсилає інформацію до браузера.
- 4) JavaScript отримує відповідь, розшифровує її та виводить користувачу.

Для обміну даними між JS і PHP (рис. 3.1.2) на сторінці створюється об'єкт XMLHttpRequest, який виконує функцію посередника між браузером та сервером. Запити можуть надсилатися в одному з двох типів – GET і POST. У першому випадку звернення проводиться до документа на сервері, у ролі аргументу йому передається URL сайту. Для запобігання перериванню запиту можна скористатися функцією JavaScript Escape. Для великих обсягів даних використовується функція POST.

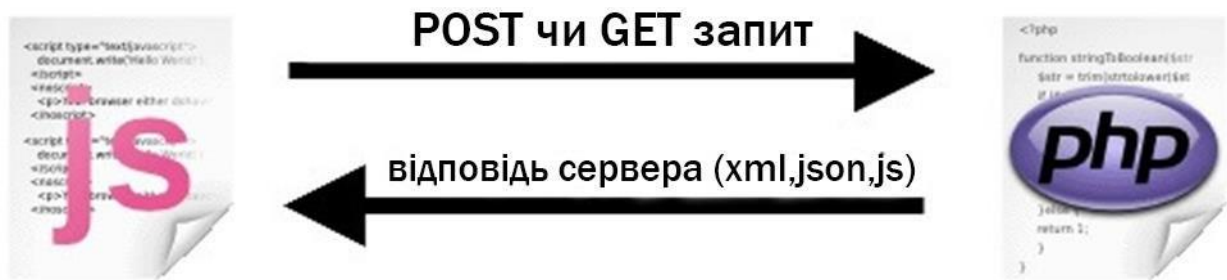


Рисунок 3.1.2 – Обмін даними між JS і PHP

Сторона сервера обробляє вхідні дані і на їх основі створює нову інформацію, яка буде надіслана клієнту.

AJAX застосовує асинхронну передачу даних. Такий підхід дозволяє користувачеві робити різні дії під час «фонового» обміну інформації з сервером. Діє сповіщення користувача про процеси, що протікають, щоб він не подумав, що сайт «завис» або на ньому стався якийсь збій.

Як відповідь сервер використовує простий текст, XML та JSON. У першому випадку результат можна відразу відобразити на сторінці. При отриманні документа XML його зазвичай конвертують в HTML і виводять на екран. Якщо відповідь отримана у форматі JSON, клієнту слід виконати отриманий код. Після цього буде сформовано об'єкт JavaScript.

До переваг технології AJAX можна віднести:

1. Скорочення трафіку. Обсяг даних під час роботи з web-додатками значно знижується. Це відбувається за рахунок того, що не потрібно завантажувати всю сторінку цілком, достатньо отримати лише змінену частину або набір даних. Після цього JavaScript змінює вміст сторінки у браузері.

2. Зниження навантаження на сервер. Грамотне використання AJAX дозволяє багаторазово зменшити навантаження на сервер. Наприклад, можна використовувати шаблон для створення постійних елементів сайту: логотипу, меню тощо. Крім того, для задоволення конкретного запиту не потрібно оновлювати всю сторінку. Наприклад, при голосуванні на сайті користувач вибирає потрібний пункт, натискає кнопку, інформація надсилається на сервер,

після чого надходить відповідь. Протягом усього часу оновлення сторінки не відбувається.

3. Збільшення швидкості роботи сервісу. Оскільки підвантажується лише змістовна частина, користувач набагато швидше бачить результат своїх дій. Широкий спектр можливостей. Використання AJAX не обмежується формами. Наприклад, при проходженні реєстрації на деяких сервісах користувач вводить логін – і за мить йому видається інформація про те, вільний він чи ні. Також після введення пошукового запиту в Google після кожної літери чи слова пропонується кілька варіантів запиту. Це значно підвищує комфорт роботи із сайтами.

Серед недоліків AJAX варто виділити наступні:

1. JavaScript має бути увімкнено. Сторінки сайтів, створені за технологією AJAX, не можуть нормально працювати за відключеної підтримки JavaScript. На них не можна розмістити закладки, та й пошукові системи далеко не завжди можуть їх проіндексувати.

2. Неможливість інтеграції із інструментами браузера. При динамічному формуванні сторінок браузер не відображає їх в історії відвідування, тому кнопка «Назад» не допоможе переміститися на попередній етап роботи. Втім, таку проблему можна вирішити за рахунок спеціальних скриптів. Також відсутня можливість встановити на потрібний матеріал закладки.

3. Погіршення безпеки. Помітним недоліком AJAX також є прогалини у безпеці, адже вихідний код кожен може прочитати у браузері.

4. Неможливість встановити кількість звернень. Механізм динамічного підвантаження контенту помітно спотворює дані статистики. Адже в цьому випадку при переміщенні користувача різними сторінками не виконується операція їх перезавантаження, і звичайний лічильник не реєструє перехід. Для великих проектів таке штучне заниження кількості переглядів призводить до значного падіння доходів.

Проблеми в індексації контенту пошуковими роботами. Часто зміст, який завантажується на сторінки динамічним способом, виявляється недоступним для роботів. Тому фахівці рекомендують використовувати динамічне завантаження

для певних частин контенту. У цьому негативний вплив AJAX на SEO можна мінімізувати.

Сайти з технологією AJAX можуть мати гіршу репутацію пошукових систем у порівнянні з аналогічними ресурсами без її використання. Серед основних причин виділяють:

- можливість, що пошукові роботи не врахують весь контент;
- марний адресний рядок (у всіх сторінок однакова адреса);
- робот може бачити не той зміст сторінки, що відображається користувачеві.

Цих негативних сторін можна уникнути, якщо використовувати AJAX за цільовим призначенням для динамічної взаємодії з сервером. Наприклад, частину статті з ключовими словами встановити нединамічно на початку сторінки.

Динамічні сторінки можна кешувати та відображати їх як статичні. Для виклику AJAX краще скористатися класичним якорем, ніж подією onClick.

Таким чином, грамотне використання AJAX дозволить підвищити швидкість роботи сайту та його зручність для користувачів, не жертвуючи дружнім ставленням пошукових систем.

В якості сервера для коректної роботи сайту і бази даних було обрано Open Sever Panel.

Open Server Panel – це портативне програмне середовище, створене спеціально для веб-розробників з урахуванням їх рекомендацій та побажань.

Даний програмний комплекс включає ретельно підібраний набір серверного програмного забезпечення, а так само неймовірно зручну і продуману керуючу утиліту, яка володіє потужними можливостями з адміністрування і налаштування всіх доступних компонентів.

Можливості програми:

- швидкі старт та зупинка;
- автостарт сервера під час запуску програми;
- декілька режимів управління доменами;

- монтування віртуального диска;
- підтримка керування через командний рядок;
- підтримка профілів налаштувань;
- зручний перегляд логів усіх компонентів;
- перемикання HTTP, MySQL та PHP модулів;
- детальна та зрозуміла документація;
- швидкий доступ до шаблонів конфігурації;
- інтерфейс на декількох мовах;
- автозапуск програм за списком.

OSPanel широко використовується з метою розробки, налагодження та тестування веб-проектів, а також для надання веб-сервісів у локальних мережах.

Особливості середовища:

- не потребує встановлення (портативність);
- одночасна робота з Denwer, Xampp тощо;
- робота на локальному/мережевому/зовнішньому IP;
- підтримка SSL без будь-якої додаткової. налаштування;
- створення домену шляхом створення звичайної папки;
- підтримка кириличних доменів;
- підтримка аліасів (доменних покажчиків);
- захист сервера від зовнішнього доступу;
- runuscode конвертер доменних імен;
- набір популярних сторонніх розширень PHP;
- планувальник завдань (cron);

В якості системи адміністрування базою даних був використаний PhpMyAdmin.

PhpMyAdmin. Під час розробки веб-ресурсів чи перенесення сайтів необхідно отримати доступ до БД – бази даних проектів на сервері. Найпростіший метод – командний рядок. Інший варіант – графічний інтерфейс, що допомагає подати дані у вигляді ієрархічних списків. PhpMyAdmin – популярний інструмент для роботи з БД, має відкритий код, написаний мовою

PHP. Працює через браузер, може бути інтегрований в інтерфейс програмного забезпечення або панель управління хостингом.

PhpMyAdmin – це зручне робоче середовище для новачків та професіоналів. Інтуїтивно зрозуміле розташування основних елементів допомагає швидко отримати доступ до баз даних. Інтерфейс перекладено кількома мовами, включаючи українську. Вільна ліцензія сприяла поширенню програми по всьому світу. Продукт інтегрований у Openserver, Денвер та інші розробки. Перша версія з'явилася 1998 року, розробник Тобіас Ротшильд. Підтримується на всіх операційних системах. Формати файлів, що читаються: xml, csv, json, pdf, xml. Починаючи з третьої версії, для роботи потрібно PHP 5.2.

За допомогою графічного інтерфейсу можна шукати дані в базі, виконувати SQL-команди, додавати нових користувачів та наділяти їх правами адміністратора, змінювати ім'я облікові записи та паролі, коригувати наявні записи.

Робочий простір поділено на три зони. У колонці зліва розташована база даних сайту, подана у вигляді деревоподібної структури. Навігація здійснюється за допомогою пошукового рядка або шляхом переходу від верхнього рівня до нижнього. У налаштуваннях користувач вибирає умови угруповання пунктів дерева, спосіб відображення таблиці. На рис. 3.1.3 представлено вікно робочого простору PhpMyAdmin.

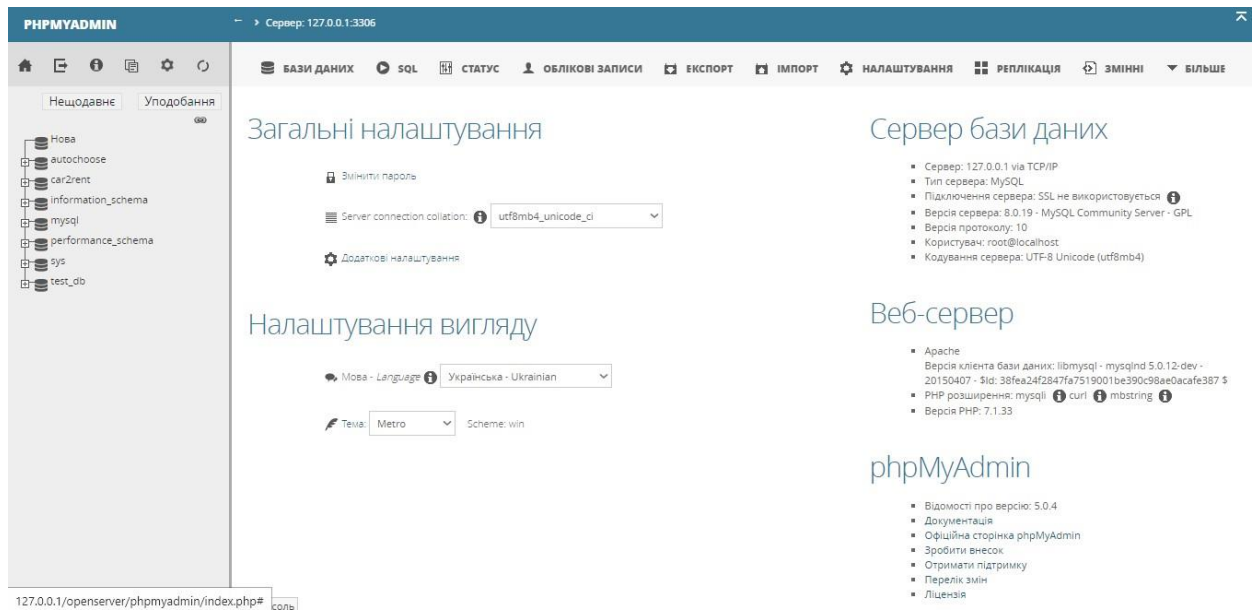


Рисунок 3.1.3 – Вікно робочого простору PhpMyAdmin

Слід зазначити, що у правій частині екрана вказується сервер бази даних, web-сервіс та інструкція з роботи з PhpMyAdmin. Над центральною частиною сторінки розташоване головне меню. З його допомогою налаштовується зіставлення кодувань, зовнішній вигляд робочого простору: мова, тема, розмір шрифту. Інтерфейс дозволяє імпортувати або експортувати файли, одним кліком відновлювати налаштування користувача за замовчуванням. Вибравши таблицю у бічному меню, можна додати нового користувача, встановити привілеї доступу. Крім таблиць, у робочому середовищі є ще два вікна. У першому виконуються SQL-запити на сервер, у другому відображається використаний мережевий трафік з моменту запуску.

PhpMyAdmin – це основний інструмент роботи з динамічними сайтами. Щоб сформувати контент, необхідно створити HTML-каркас та відобразити на сторінці за допомогою PHP скриптів інформацію з бази даних. Першою генерується користувальницька база даних з привілеями адміністратора: можна створювати, видаляти, редагувати таблиці, додавати нові рядки. Інтуїтивно зрозумілий інтерфейс не вимагає знання мови, достатньо розуміти синтаксис: Type, Table, Alter, Create. Потрібний рядок можна знайти за допомогою швидкого пошуку.

3.2 Проектування структури бази даних.

Створення бази даних слід починати з її проектування (розробки). У результаті проектування має бути визначена структура бази, тобто склад таблиць, їхня структура та логічні зв'язки. Структура реляційної таблиці визначається складом стовпців, їхньою послідовністю, типом даних кожного стовпця та їхнім розміром, а також ключем таблиці. Процес проектування можна здійснювати двома підходами.

За першого підходу спочатку визначають основні задачі, для розв'язання яких створюється база, та потреби цих задач у даних.

За другого підходу визначають предметну область (сферу), здійснюють аналіз її даних і встановлюють типові об'єкти предметної області. Найбільш раціональним підходом проектування бази даних є поєднання обох підходів.

Як відомо, етап проектування бази даних (БД) вважається одним найскладніших етапів проектування. Порівняно з аналізом вимог до БД або розробкою додатків, проектування БД, на думку багатьох провідних фахівців, є невдало структурованим завданням. Якщо всі етапи створення БД перекриваються один з одним у своїй послідовності, то етап проектування перекривається з усіма іншими етапами.

Під час розробки сайту «AUTOCHOOSE» було створено дев'ять таблиць: announcement, body, brand, drive_type, fuel, gearbox, model, uploaded_img, users.

Таблиці: body, brand, drive_type, fuel, gearbox, model, users зв'язані з головною таблицею announcement (рис. 3.2.1).

Розглянемо короткий опис назв таблиць та їх призначення.

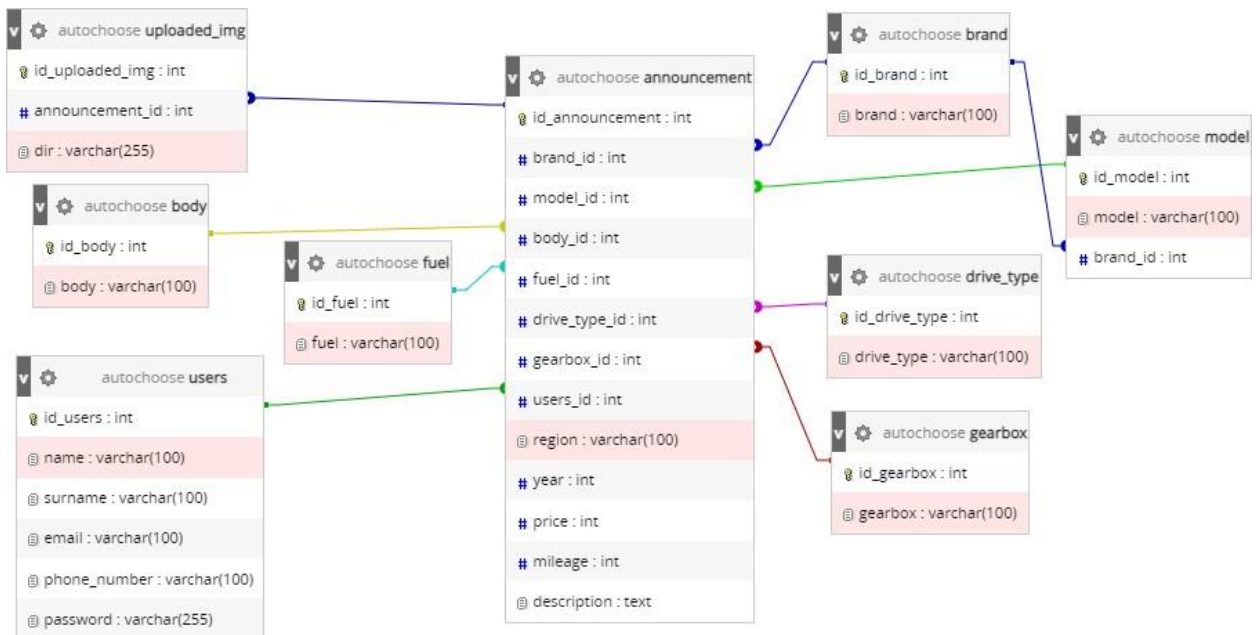


Рисунок 3.2.1 – Структура бази даних

1. announcement – таблиця створена для збереження всіх оголошень;
2. body – таблиця, створена для повного опису типу кузова автомобіля;
3. brand – таблиця, створена для збереження всіх марок автомобіля;
4. drive_type – таблиця, створена для збереження інформації про тип приводу автомобіля;
5. fuel – таблиця, створена для збереження інформації про тип двигуна автомобіля;
6. gearbox – таблиця, створена для збереження інформації про тип коробки передач автомобіля;
7. model – таблиця, створена для збереження інформації про модель автомобіля;
8. uploaded_img – таблиця інформації для збереження завантажених фотографій автомобіля;
9. users – таблиця, створена для збереження інформації про зареєстрованих користувачів.

Таблиця announcement містить в собі повну інформацію про кожний автомобіль, а також деякі стовпці цієї таблиці об'єднані з іншими таблицями зовнішніми ключами для дотримування правил створення реляційної бази даних.

В таблиці 3.2.1 представлено повний опис оголошення.

Таблиця 3.2.1 – Таблиця повного опису оголошення

Властивість	Тип	Розмірність	Ідентифікатор
Ідентифікатор	int	11	id
Марка авто	int	11	brand_id
Модель авто	int	11	model_id
Тип кузова	int	11	body_id
Тип двигуна	int	11	fuel_id
Тип приводу	int	11	drive_type_id
Тип коробки передач	int	11	gearbox_id
Власник авто	int	11	users_id
Регіон	varchar	100	region
Рік випуску	int	11	year
Ціна	int	11	price
Пробіг	int	11	mileage
Опис	text	—	description

Розглядаючи таблицю body, слід зазначити, що деякі стовпці таблиці об'єднані з основною таблицею зовнішнім ключем для дотримування правил створення реляційної бази даних.

В таблиці 3.2.2 представлено повну інформацію про тип кузова автомобіля.

Таблиця 3.2.2 – Таблиця повного опису типу кузова автомобіля

Властивість	Тип	Розмірність	Ідентифікатор
Ідентифікатор	int	11	id
Тип кузова	varchar	100	body

Оскільки таблиця brand створена для збереження всіх марок автомобіля, то деякі стовпці цієї таблиці об'єднані з основною таблицею зовнішнім ключем для дотримувannya правил створення реляційної бази даних.

В таблиці 3.2.3 представлено повну інформацію про марки автомобіля.

Таблиця 3.2.3 – Таблиця повного опису марок автомобіля

Властивість	Тип	Розмірність	Ідентифікатор
Ідентифікатор	int	11	id
Марка авто	varchar	100	brand

Розглядаючи таблицю drive_type бачимо, що деякі стовпці цієї таблиці об'єднані з основною таблицею зовнішнім ключем для дотримувannya правил створення реляційної бази даних.

В таблиці 3.2.4 представлено повну інформацію про тип приводу автомобіля.

Таблиця 3.2.4 – Таблиця повного опису типу приводу автомобіля

Властивість	Тип	Розмірність	Ідентифікатор
Ідентифікатор	int	11	id
Тип приводу	varchar	100	drive

В таблиці fuel також деякі стовпці цієї об'єднані з основною таблицею зовнішнім ключем для дотримувannya правил створення реляційної бази даних.

В таблиці 3.2.5 представлено повну інформацію про тип двигуна автомобіля.

Таблиця 3.2.5 – Таблиця повного опису типу двигуна автомобіля

Властивість	Тип	Розмірність	Ідентифікатор
Ідентифікатор	int	11	id
Тип двигуна	varchar	100	fuel

Розглядаючи таблицю gearbox, видно, що деякі стовпці таблиці gearbox об'єднані з основною таблицею зовнішнім ключем для дотримування правил створення реляційної бази даних.

В таблиці 3.2.6 представлено повну інформацію про тип коробки передач.

Таблиця 3.2.6 – Таблиця повного опису типу коробки передач

Властивість	Тип	Розмірність	Ідентифікатор
Ідентифікатор	int	11	id
Тип коробки передач	varchar	100	gearbox

Як і в розглянутих вище таблицях, деякі стовпці таблиці model об'єднані з основною таблицею та таблицею brand зовнішнім ключем для дотримування правил створення реляційної бази даних.

В таблиці 3.2.7 представлено повну інформацію про модель автомобіля.

Таблиця 3.2.7 – Таблиця повного опису моделі автомобіля

Властивість	Тип	Розмірність	Ідентифікатор
Ідентифікатор	int	11	id
Модель автомобіля	varchar	100	model
Бренд автомобіля	int	11	brand_id

Деякі стовпці таблиці uploaded_img об'єднані з основною таблицею зовнішнім ключем для дотримування правил створення реляційної бази даних.

В таблиці 3.2.8 представлено повну інформацію про завантажені фотографії автомобіля.

Таблиця 3.2.8 – Таблиця повного опису завантажених фотографій автомобіля

Властивість	Тип	Розмірність	Ідентифікатор
Ідентифікатор	int	11	id
Номер оголошення	int	11	announcement_id
Директорія зберігання фотографії	varchar	255	dir

Таблиця users створена для збереження інформації про зареєстрованих користувачів, також деякі стовпці цієї таблиці об'єднані з основною таблицею зовнішнім ключем для дотримування правил створення реляційної бази даних.

В таблиці 3.2.9 представлено повну інформацію про зареєстрованих користувачів.

Таблиця 3.2.9 – Таблиця повного опису зареєстрованих користувачів

Властивість	Тип	Розмірність	Ідентифікатор
Ідентифікатор	int	11	id
Ім'я користувача	varchar	100	name
Прізвище користувача	varchar	100	surname
Е-mail користувача	varchar	100	email
Номер телефону	varchar	100	phone_number
Пароль користувача	varchar	255	password

3.3 Опис процесу проектування системи з пошуку та підбору транспортного засобу.

Світ навколо нас сповнений системами – механізмами, живими істотами, спільнотами людей, тобто системами механічними, біологічними, соціальними тощо. У широкому розумінні інформаційною можна назвати будь-яку систему опрацювання інформації. Інформаційна система забезпечує приймання інформації, її перетворення, опрацювання, збереження і передачу результатів опрацювання споживачу: людині, машині, іншій інформаційній системі. Прикладом сучасної інформаційної системи може бути редакція газети або журналу, оснащена комп'ютерною технікою чи інформаційна система котра спрощує пошук житла.

Інформаційна система – це сукупність засобів збору, зберігання, передачі, оброблення інформації в певній предметній області для досягнення поставленої мети у процесі управління. Інформаційна система (у загальному розумінні) – це система, яка здійснює або в якій відбуваються інформаційні процеси: пошук, збирання, зберігання, передавання й опрацювання інформації.

В інформаційній системі можуть відбуватися одночасно один, два чи кілька процесів.

Опрацювання інформації залежить від змісту вхідної інформації, але під час самого опрацювання інформація не осмислюється, а лише перетворюється згідно з попередньо розробленими алгоритмами.

Загалом інформаційна система складається з таких 3 частин:

Фронтенд – веб-інтерфейс, за допомогою якого користувач взаємодіє з системою.

Бекенд – сервер, що приймає запити від клієнтської частини, робить запит до бази даних та та вертає результат користувачу.

Сервер бази даних – комунікує з бекенд частиною та відповідає за збереження, читання, інформації.

На рис. 3.3.1 представлена архітектура сайту «AUTOCHOOSE».

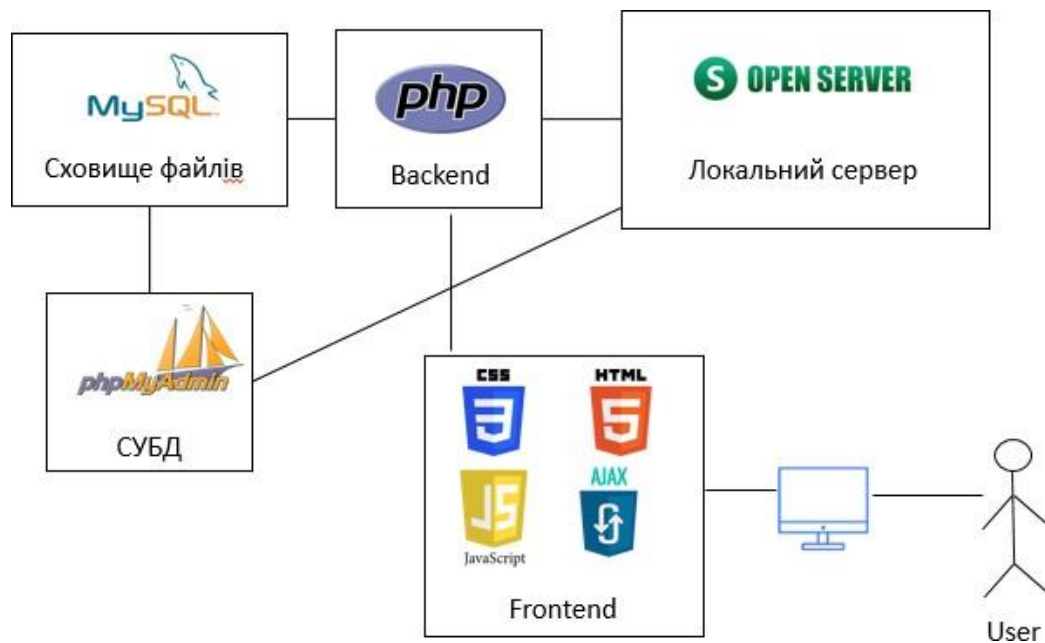


Рисунок 3.3.1 – Архітектура сайту «AUTOCHOOSE»

3.3.1 Опис предметної області. Найважливішим етапом процесу розробки програмного забезпечення є етап системного аналізу і моделювання діяльності підприємства-замовника. Від успіху проведення цього етапу залежить успіх проектування в цілому.

Перш, ніж представити систему з пошуку та підбору транспортного засобу «AUTOCHOOSE» у вигляді веб-сайту, необхідно описати предметну область.

Предметна область являє собою приватне підприємство сфери автопослуг «AUTOCHOOSE». ПП «AUTOCHOOSE» планується реалізувати в місті Чорнобай Черкаської області у 2024 р. Воно буде надавати такі консультаційні послуги:

- якісної компетентної консультації з продажу та підбору транспортного засобу;
- передпродажна підготовка;
- перевірка транспортного засобу перед покупкою;
- здійснення юридичного супроводу операцій з придбання транспортного засобу.

Всі працівники підприємства «AUTOCHOOSE» будуть мати сертифікат спеціаліста з питань авто, швидко та якісно зможуть надати інформацію та консультацію користувачу.

3.3.2 Формування вимог до системи. Нефункціональні вимоги описують «як» повинна працювати система або програмний продукт та які властивості або характеристики повинна мати. Як правило, говорячи про нефункціональні вимоги, найчастіше говорять про атрибути якості (тобто вимоги, що визначають якісні характеристики розроблюваного ПЗ або системи – продуктивність, надійність, масштабованість).

Виділимо 2 атрибути якості, а саме: продуктивність (performance) та зручність використання (usability). Система, що розробляється в курсовій роботі, пов'язана з автоматизацією з пошуку та підбору транспортних засобів.

Отже, виходячи з цього, вона повинна виконувати задані функції з певним рівнем швидкодії, наприклад, обробляти запити до серверної частини в діапазоні від 300 до 500 мс. Також система повинна забезпечити зручність використання для початкового навчання та ефективної роботи, і для подальшої її експлуатації.

Для сформованих функціональних та нефункціональних вимог процесу в роботі розроблена діаграма вимог у нотації UML (рис. 3.3.2.1).

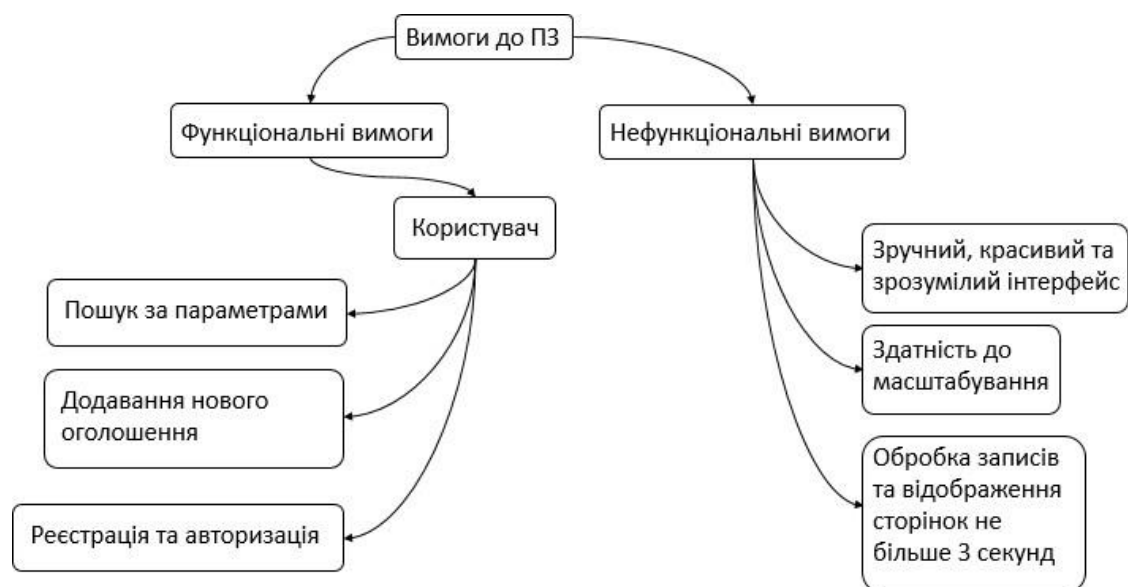


Рисунок 3.3.2.1 – Діаграма вимог до розроблюваної системи

Перед розробкою вимог необхідно створити спочатку систему бізнес-правил.

Система бізнес-правил – логічний опис стандартів, процедур і образу дій, якими керується підприємство у своїй роботі.

Бізнес-правила використовуються для визначення взаємозв'язків між сутностями, діями, які можуть виконуватися над сутностями, проміжних контрольних точок і робочих процесів.

Бізнес-правила також описують обмеження на образ дій, які повинні враховуватися системою. Правила слід складати так, щоб вони були зрозумілі як керівникам системи, так і розробникам, незалежно від мети або мови програмування.

Під час аналізу предметної області була розроблена наступна система бізнес-правил:

- користувачі в системі характеризуються ім'ям, прізвищем, електронною поштою;
- в системі не повинно існувати користувачів з однаковою поштою;
- вхід до системи виконується через пароль та адресу електронної пошти;
- додавання нових оголошень дозволено тільки зареєстрованим користувачам.

3.3.3 Проектування інтерфейсу користувача. Інтерфейс – це механізм, що забезпечує взаємодію користувача з програмним продуктом. Він, насамперед, повинен враховувати потреби кінцевих користувачів. Під час аналізу аналогічних систем було створено приблизні мокапи для зручного відображення інформації та інтуїтивно зрозумілого інтерфейсу сервісу.

Акцент при розробці дизайну був зроблений на компактність подачі інформації, використання таблиць, мінімалістичний інтерфейс, що робить веб-застосування комфортним для роботи (рис. 3.3.3.1-3.3.3.2).

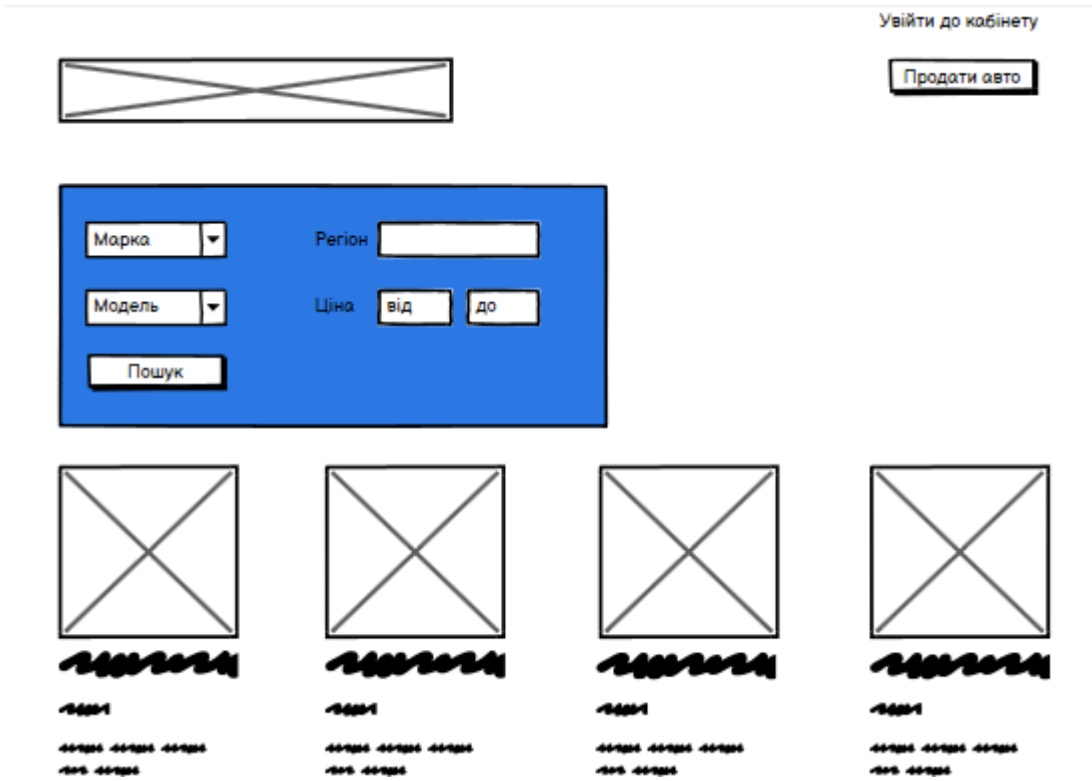


Рисунок 3.3.3.1 – Макет головної сторінки веб-сайту «AUTOCHOOSE»

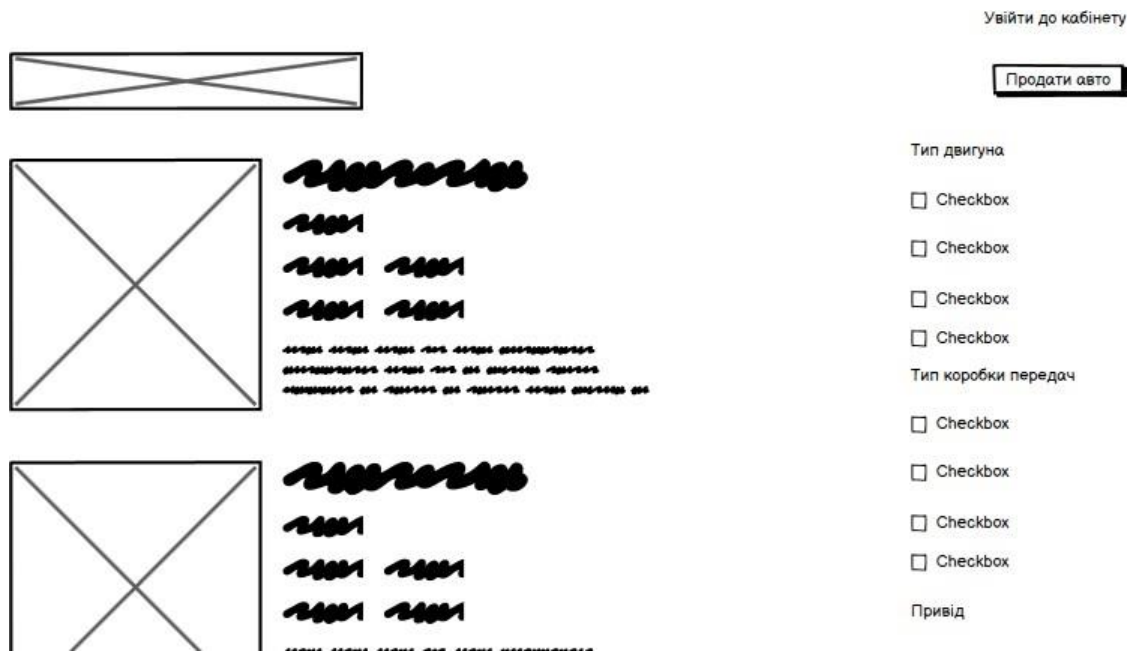


Рисунок 3.3.3.2 – Макет сторінки оголошень

3.3.4 Проектування інформаційних систем з пошуку та підбору транспортного засобу.

Розробка інформаційних систем включає в себе кілька етапів. Однак, завжди початковим етапом створення системи є вивчення, аналіз та моделювання діяльності замовника для можливого поліпшення і оптимальних методів роботи, які і будуть

реалізовані в створюваному додатку. Для цього доцільно скористатися можливостями засобу AllFusion Process Modeler (BPwin).

AllFusion Process Modeler (раніше BPwin) – інструмент, призначений для моделювання, аналізу, документування та оптимізації бізнес-процесів і має достатньо простий та інтуїтивно зрозумілий інтерфейс користувача, що дає можливість аналітикові створювати складні моделі при мінімальних зусиллях. Він призначений для побудови контекстної діаграми бізнес-процесу з метою полегшення праці і збільшення продуктивності системного аналітика на першому етапі розробки системи.

IDEF0 – це методологія графічного опису систем і процесів діяльності організації як безлічі взаємозалежних функцій. Особливістю IDEF0 є її акцент на ієрархічне представлення об'єктів, що значно полегшує розуміння предметної області. Вона дозволяє досліджувати функції організації, не пов'язуючи їх з об'єктами, що забезпечують їх реалізацію. В IDEF0 розглядаються логічні зв'язки між роботами, а не послідовність їх виконання в часі.

Перед безпосередньою розробкою програмного продукту для кращого розуміння та аналізу бізнес-процесів засобами AllFusion Process Modeler, потрібно побудувати два типи схем: IDEF0 та DFD. Перш за все, перед побудовою IDEF0-діаграми сформулюємо задачу: показати процес додавання оголошення та пошуку оголошень.

У процесі аналізу предметної області складена контекстна діаграма моделі у стандарті IDEF0 засобами AllFusion Process Modeler (рис. 3.5.1).

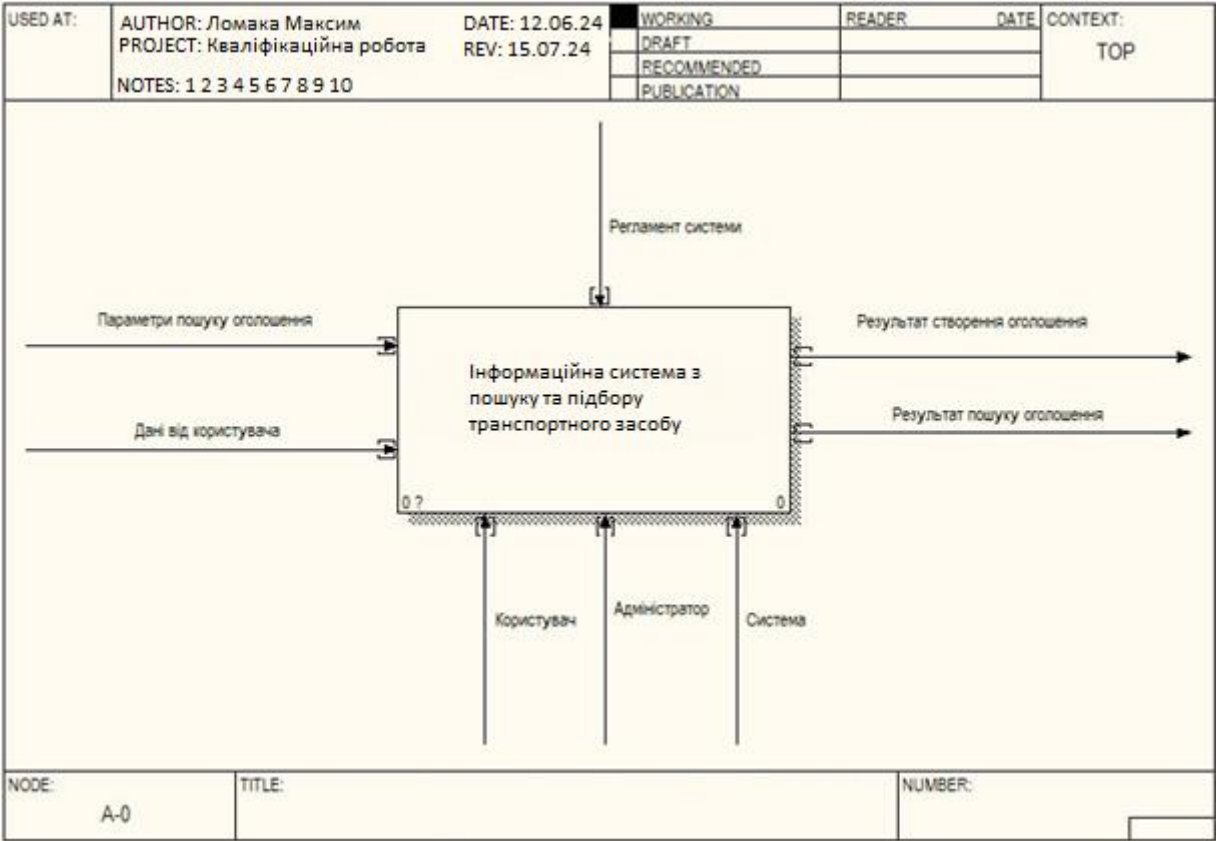


Рисунок 3.3.4.1 – Контекстна діаграма моделі у стандарті IDEF0

Декомпозиція контекстної діаграми (рис. 3.3.4.2) реалізована на виділенні наступних робіт: знаходження нових оголошень, зміна фільтрів, обробка запиту клієнта. Таким чином, була побудована наступна діаграма 1-го рівня декомпозиції.

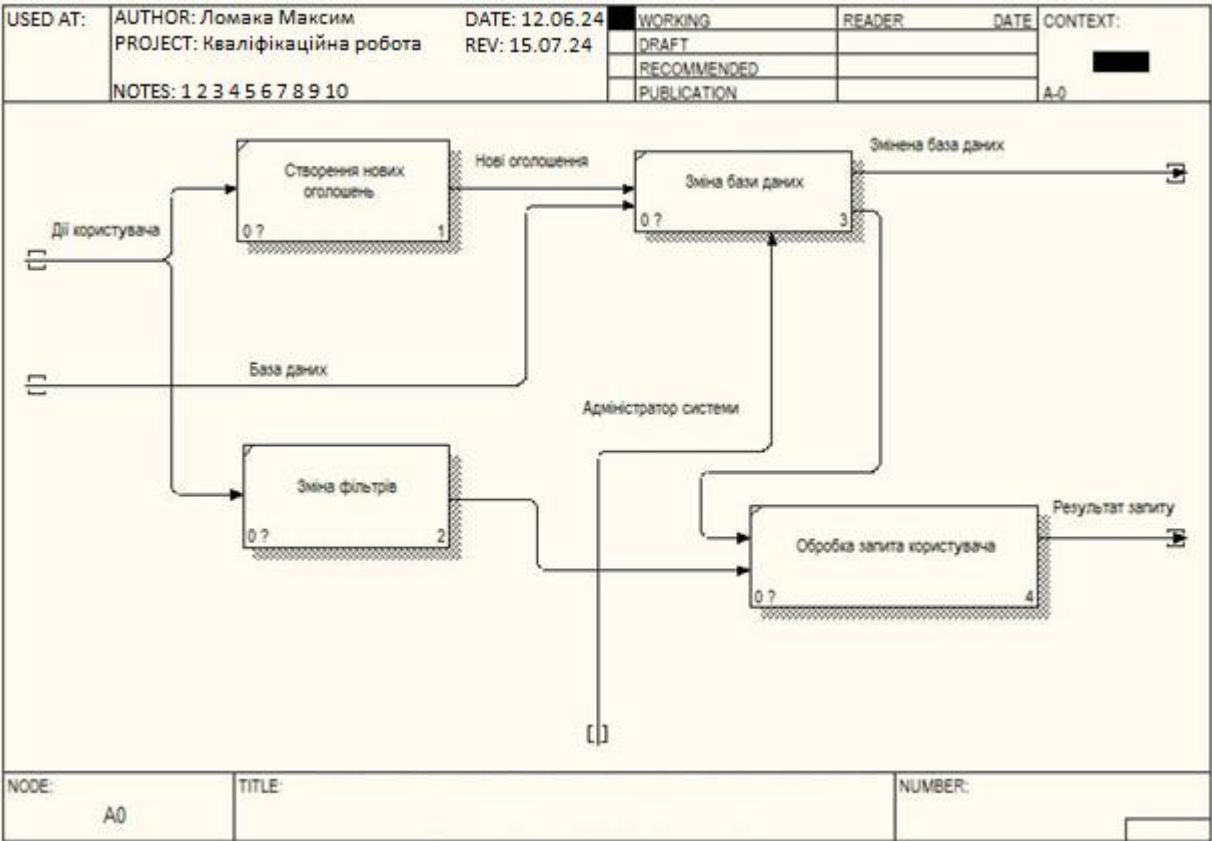


Рисунок 3.3.4.2 – Декомпозиція контекстної діаграми у стандарті IDEF0

Робота 1-го рівня декомпозиції має такі інтерфейсні дуги:

- вхід: параметри пошуку оголошення, БД оголошень, дія користувача;
- вихід: результат запиту, змінена БД;
- механізми: адміністратор, користувач.

3.4 Розробка структури системи «AUTOCHOOSE»

Провівши аналіз сайтів існуючих сервісів системи з пошуку та підбору транспортного засобу, визначено, що буде доречним виконати меню розроблюваної системи «AUTOCHOOSE», що складається з таких розділів:

Головна сторінка:

- Фільтр пошуку оголошень;
- Створити нове оголошення;
- Профіль користувача;
- Реєстрація;
- Авторизація.

Структура системи у вигляді сайту «AUTOCHOOSE» представлена на рисунку 3.4.1.

Вона має плоску структуру: до всіх розділів сайту можна отримати доступ з основного навігаційного меню.



Рисунок 3.4.1 – Структура сайту «AUTOCHOOSE»

В цьому розділі представлений опис функціонального програмного середовища та обґрунтовано вибір сучасних технологій web-програмування для проектування і розробки інформаційних систем з пошуку та підбору транспортного засобу «AUTOCHOOSE».

Описано предметну область та представлено проектування інформаційних систем з пошуку та підбору транспортного засобу за допомогою засобу моделювання AllFusion Process Modeler. Створено базу даних, структуру та представлено головну сторінку системи «AUTOCHOOSE».

3.5 Тестування то дослідницька експлуатація

3.5.1 Особливості тестування веб-орієнтованого програмного забезпечення

Так, як сьогодні веб-сайти мають важливе значення для будь-якого бізнесу, питання правильного тестування сайтів заслуговує все більшої уваги з боку компаній. Тестування сайтів допомагає розробнику виявити слабкі місця проекту, і виправити їх для того, щоб взаємодія користувача з сайтом була максимально комфортною.

Тестування – це перевірка сайту, що спрямована на виявлення відхилень, тобто процес пошуку помилок (багів). Перевірку можуть проводити як виконавець, так і замовник. Але найефективнішою буде спільна робота.

Розглянемо основні етапи тестування:

- ознайомлення із завданнями, документацією та макетами;
- складання плану подальшої роботи;
- перевірка описаного функціоналу, тобто функціональне тестування. Воно полягає у перевірці: роботи пошуку; всіх основних функцій сайту; так званої працездатності сайту (наприклад: додавання, видалення, редагування, відображення, збереження оцінок).

Основними недоліками такого тестування є: ймовірності недостатності роботи над логічними помилками та надмірного тестування.

Тестування верстки. Спочатку тестувальнику необхідно перевірити правильність розташування елементів та їх відповідність макетам. Після цього проводиться перевірка коду на валідність – це відповідність вихідного коду правилам та нормам. Перевагами цієї перевірки є вища швидкість завантаження та висока кросбраузерність.

Для перегляду інтернет-сайтів користувачі мережі Інтернет використовують спеціальні програми – браузері. Ще кілька років тому відвідувачам інтернету були відомі лише один-два браузери, але тепер число таких програм обчислюється десятками, особливо, якщо враховувати, що кожна з них має по декілька версій, що відрізняються по функціональності.

У зв'язку з цим з'явилося поняття кросбраузерності – поняття, що описує властивість сайту ідентично працювати і відображатися у всіх браузерах (зазвичай враховуються лише найбільш поширені). Під ідентичністю розуміється відсутність розвалів верстки і здатність відображати матеріал з однаковим ступенем читабельності.

Поняття «кросбраузерність» дуже часто плутають з попіксельно відповідністю, що насправді є різними поняттями.

Отже, кросбраузерність сайту – це здатність сайту правильно і в повній мірі відображатися у всіх браузерах.

Кросбраузерність дає змогу сайту відображатись і відповідно працювати у найпоширеніших браузерах (Internet Explorer, Opera, Firefox, мобільні браузери тощо). Цей етап перевірки є досить важливим, адже досить часто з'являються нові версії різноманітних браузерів, до яких необхідно «пристосовуватись», щоб кожен користувач зміг ознайомитись вашим сайтом.

Usability тестування – цей процес включає в себе перевірку програмного продукту на зручність, аналіз контенту, оцінку структури посилань, відповідність до заданого дизайну.

Досить цікавим та більш ефективним є те, що в ньому задіяні як самі тестувальники, так і користувачі (від 4 до 6 людей).

Тестування безпеки (Security testing) – цей вид тестування необхідний аби уникнути атак, перевіряються права доступу.

Тестування безпеки необхідне для:

- різноманітних соціальних програм;
- платіжних систем;
- web-додатків;
- додатків яких міститься прихована інформація (комерційна).

Тестування продуктивності сайту (Performance testing) базується на перевірці цілісності, комплексності, працездатності та можливість ефективної роботи сайту в умовах значного навантаження.

Цей вид тестування складається із таких типів:

- стрес тестування (Stress testing);
- тестування стабільності або напруження на відмову (Stability / Reliability testing);
- шип тестування (Spike Testing);
- об'ємне тестування (Volume Testing);
- навантажувальне тестування (Load testing).

Впродовж усього процесу тестування тестувальник виявляє помилки, знаходить

шляхи їх вирішення та виправляє. Після цього створює звіт, який передається учасникам проекту та його керівнику. Необхідно знову провести повторне тестування, щоб перевірити якість виправлених недоліків та запобігти виникненню нових. Перевірка сайту проводиться до того часу, поки він не буде відповідати специфіці технічних завдань.

Якщо робота виконана якісно, то заключним етапом являється розміщення сайту на сервері. Тестування є однією з найважливіших частин будь-якої програми контролю якості.

3.5.2 Перевірка сайту на кросбраузерність

Поняття кросбраузерності – поняття, що описує властивість сайту ідентично працювати і відображатися у всіх браузерах (зазвичай враховуються лише найпоширеніші). Під ідентичністю розуміється відсутність розвалів верстки і здатність відображати матеріал з однаковим ступенем читабельності.

Проводити тестування сайту на кросбраузерність найкраще на локальному комп'ютері, на якому встановлені різні браузери, різні версії браузерів і існує можливість змінювати роздільну здатність монітора. Зробити це можна шляхом створення декількох віртуальних машин з різними ОС, на яких і будуть установлені різні браузери (Linux, Windows, Mac OS і BSD браузери). Перевірити сайт на кросбраузерність можна, встановивши на локальному комп'ютері веб-сервер (ISS, Apache) і, звернувшись до сторінки через IP-адресу комп'ютера.

Недоліки цього способу перевірки: протестувати сайт можна тільки під однією ОС і під браузерами, сумісними із цією ОС, у більшості випадків користувачу не вдасться встановити більш старі версії браузерів паралельно новим версіям. При цьому затрачається величезна кількість часу на встановлення веб-сервера і браузерів/різних версій браузерів.

Кросбраузерність дає змогу сайту відображатись і відповідно працювати у найпоширеніших браузерах (Internet Explorer, Opera, Firefox, мобільні браузери тощо). Цей етап перевірки є досить важливим, адже досить часто з'являються нові версії різноманітних браузерів, до яких необхідно «пристосовуватись», щоб кожен користувач зміг ознайомитись вашим сайтом.

Для полегшення перевірки сайту на сумісність із різними браузерами є вже готові утиліти та онлайн-сервіси. Використовуючи такі сервіси, користувач позбавить себе від

установки всіх браузерів, що перевіряються, або версій браузерів, віртуальних машин. Буде достатньо лише встановити утиліту або відвідати онлайн сервіс перевірки кросбраузерності й у результаті можна одержати скріншоти відображення сайту в різних браузерах, а після чого порівнювати та аналізувати їх.

Можна скористатися і загальносвітовою статистикою використання браузерів:

- Google Chrome (для тестування, як правило, береться остання версія), ним користуються 48,06% користувачів у світі;
- Internet Explorer, 19,6% користувачів;
- Mozilla Firefox (останньої версії), 16,74% користувачів;
- Safari, 10,63% користувачів (в США – 16,88%).

В нинішній час набирає популярності тестування кросбраузерності та кросплатформенності саме за допомогою онлайн-сервісів, таких як Saucelabs і BrowserStack.

Ключовими перевагами сервісу BrowserStack є простота, зручність, швидкість роботи, можливість тестування на справжніх платформах. Для початку тестування нічого не потрібно встановлювати чи довго налаштовувати.

3.5.3 Функціональне тестування програмного коду системи «AUTOCHOOSE»

Автоматичне тестування – це надзвичайно корисний інструмент для вбивства помилок для сучасного веб-розробника. Ви можете використовувати набір тестів, щоб вирішити або уникнути низки проблем, що розглянуті нижче.

При написанні нового коду, можна використовувати тести для перевірки правильності коду, як очікувалося.

При редагуванні або зміні старого коду, можна використовувати тести, щоб дані зміни не вплинули на поведінку програми.

Основним інструментом тестування веб-додатків є інструмент Lighthouse.

Lighthouse – це автоматизований інструмент з відкритим вихідним кодом для поліпшення якості веб-сторінок. Він проводить тестування продуктивності в браузері (емуляція швидкості з'єднання і уповільнення процесора), показує кілька метрик швидкості завантаження.

Lighthouse надає зручний спосіб поліпшення якості сайтів, дозволяючи розробникам проводити аудити продуктивності, доступності, сумісності прогресивних

веб-додатків і багато іншого. Фактично, він «допомагає не розбитися об скелі», від цього і назва – Lighthouse (з англ. – маяк).

Його можна запустити на будь-якій веб-сторінці, відкритій або тій, що вимагає аутентифікації. Він має аудит продуктивності, доступності, прогресивних веб-додатків тощо.

Категорія SEO-аудиту в Lighthouse дає розробникам і веб-майстрам можливість проводити базові SEO-перевірки стану будь-якої веб-сторінки, які знаходять потенційні можливості для оптимізації. Lighthouse працює локально в браузері Chrome, допомагаючи провести SEO-аудити сторінок в області індексування, а також на інтерактивних сторінках, публічних сторінках і сторінках, які потребують авторизації.

Можна запустити Lighthouse в Chrome DevTools, з командного рядка або як модуль Node. Потрібно надати URL-адресу Lighthouse для аудиту, і він проводить серію аудитів на сторінці, а потім генерує звіт про те, наскільки добре сторінка виконана.

Звідси використовуються неефективні аудити, як індикатори покращення сторінки. У кожному аудиті є довідковий документ, в якому пояснюється, чому важливий аудит, а також спосіб його усунення.

На рис. 3.5.1-3.5.3 представлено результати тестування сайту «AUTOCHOOSE» за допомогою інструмента для тестування веб-додатків Lighthouse.

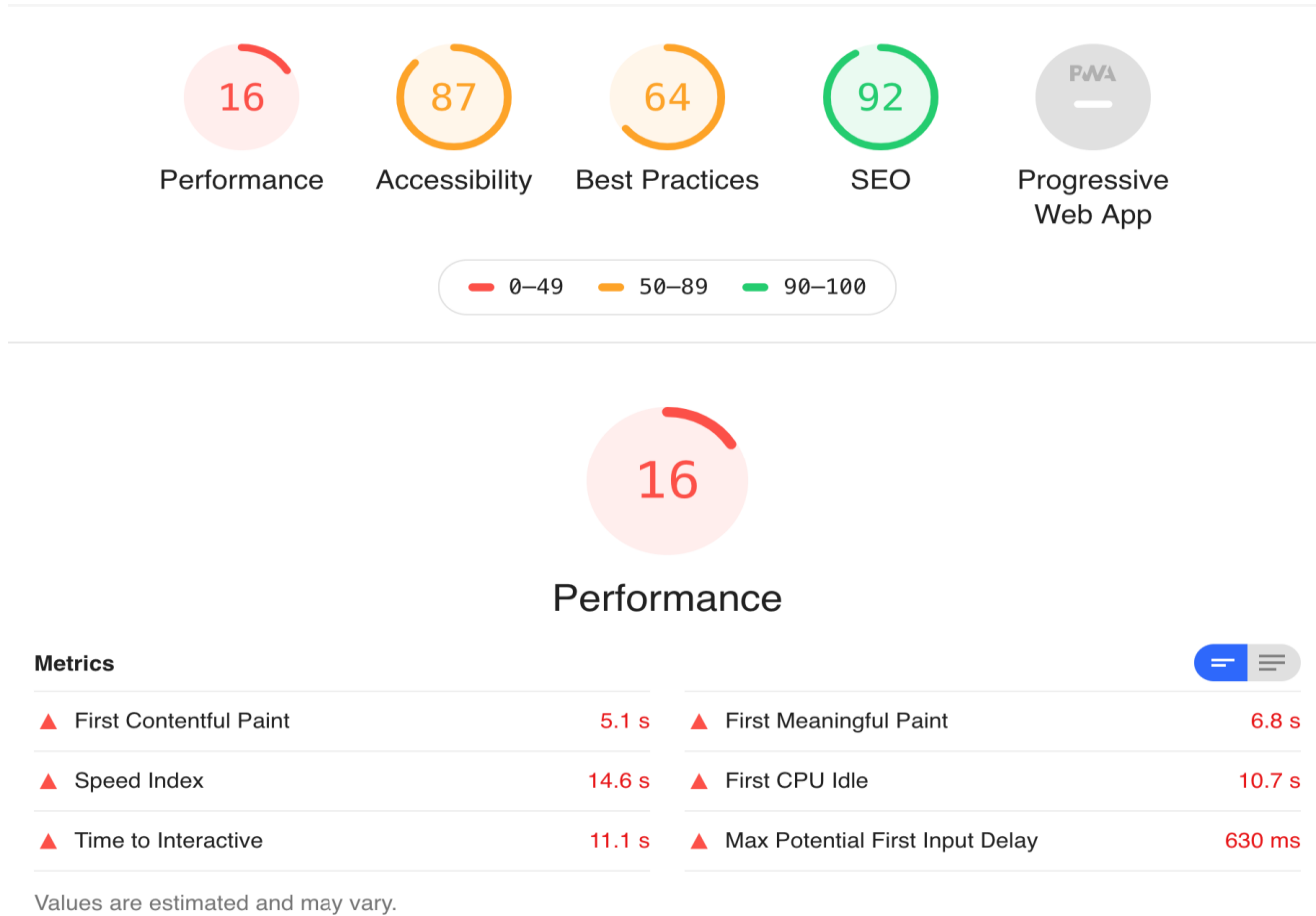


Рисунок 3.5.1 – Результат тестування сайту «AUTOCHOOSE» за допомогою Lighthouse

В ході написання кваліфікаційної роботи виконано тестування завантаження сайту «AUTOCHOOSE» (рис. 3.5.2) та якості коду сайту (рис. 3.5.3).

Opportunities — These optimizations can speed up your page load.

Opportunity	Estimated Savings
▲ Serve images in next-gen formats	3.17 s
▲ Reduce server response times (TTFB)	1.36 s
■ Defer offscreen images	0.63 s
■ Remove unused CSS	0.45 s
■ Properly size images	0.3 s
■ Preconnect to required origins	0.24 s

Diagnostics — More information about the performance of your application.

▲ Ensure text remains visible during webfont load	
▲ Serve static assets with an efficient cache policy — 23 resources found	
▲ Minimize main-thread work — 5.8 s	
■ Reduce JavaScript execution time — 2.9 s	
● Minimize Critical Requests Depth — 19 chains found	
● User Timing marks and measures — 26 user timings	
● Keep request counts low and transfer sizes small — 56 requests • 1,757 KB	

Рисунок 3.5.2 – Результат тестування завантаження сайту «AUTOCHOOSE»
за допомогою Lighthouse

**Best Practices**

▲ Does not use HTTPS — 3 insecure requests found	
▲ Uses document.write()	
▲ Links to cross-origin destinations are unsafe	
▲ Includes front-end JavaScript libraries with known security vulnerabilities — 4 vulnerabilities detected	
▲ Displays images with incorrect aspect ratio	

Рисунок 3.5.3 – Результат тестування якості коду сайту «AUTOCHOOSE»
за допомогою Lighthouse

4. Практична частина

Предметом розробки є проектування і створення системи з пошуку та підбору транспортного засобу у вигляді веб-сайту «AUTOCHOOSE».

Для реалізації проектування веб-сайту «AUTOCHOOSE» застосовувалися такі засоби:

- HTML, CSS;
- backend мова програмування: PHP;
- frontend мова програмування: JavaScript;
- асинхронна технологія AJAX;
- локальний сервер Open Server Panel;
- система управління базою даних phpMyAdmin;
- база даних: MySQL.

Основи роботи з сайтом

Для роботи з розробленою системою потрібно:

- відкрити браузер;
- ввести URL-адресу: autochoose.ua

На головній сторінці веб-сайту «AUTOCHOOSE» зображено фільтр пошуку авто за маркою, моделлю, регіоном і ціною. Трохи нижче користувач може бачити останні оголошення, які були додані на сайт (рис. 4.1).

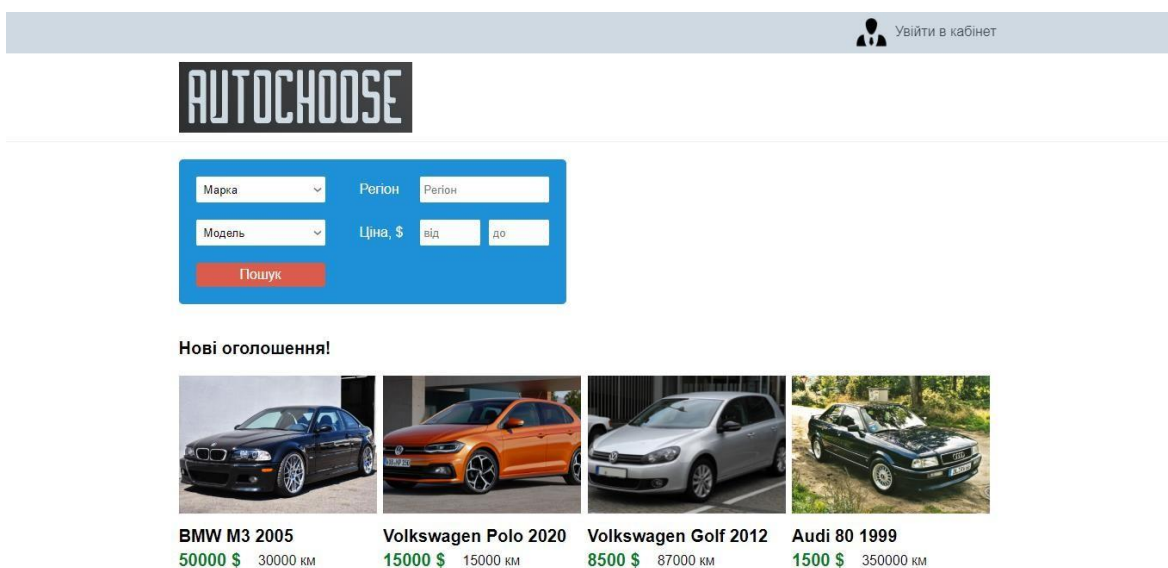


Рисунок 4.1 – Головна сторінка веб-сайту «AUTOCHOOSE»

На сторінці списку оголошень відображаються оголошення відповідно до фільтру, також доступний вибір фільтрів, які впливають на результат пошуку (рис. 4.2).

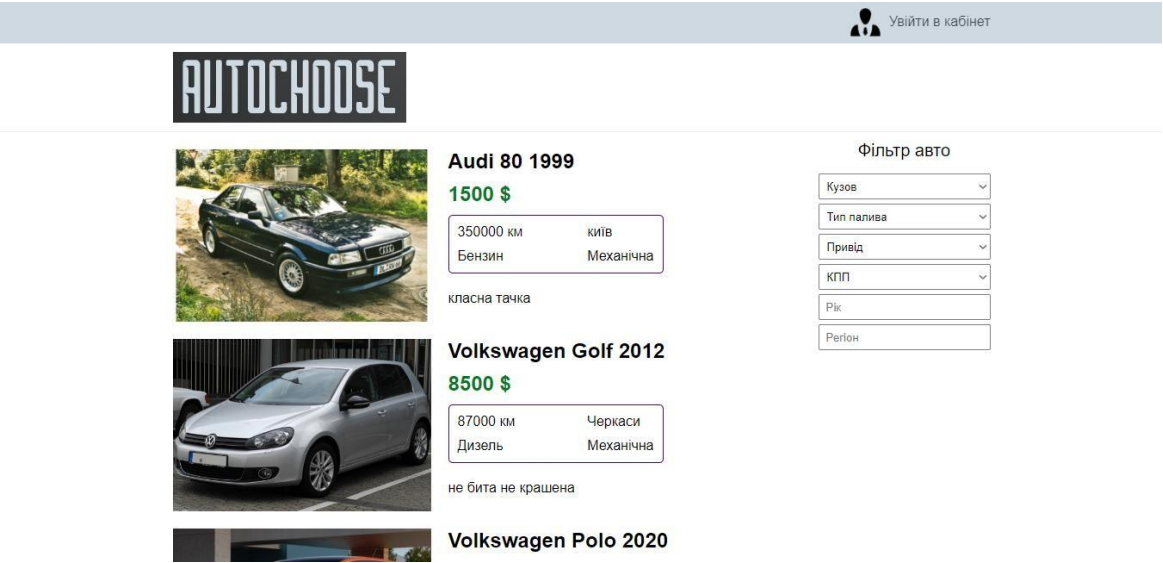


Рисунок 4.2 – Сторінка списку оголошень

На всіх сторінках сайту в нижній частині сторінки є контактна інформація, а також форма для зворотного зв'язку (рис. 4.3).

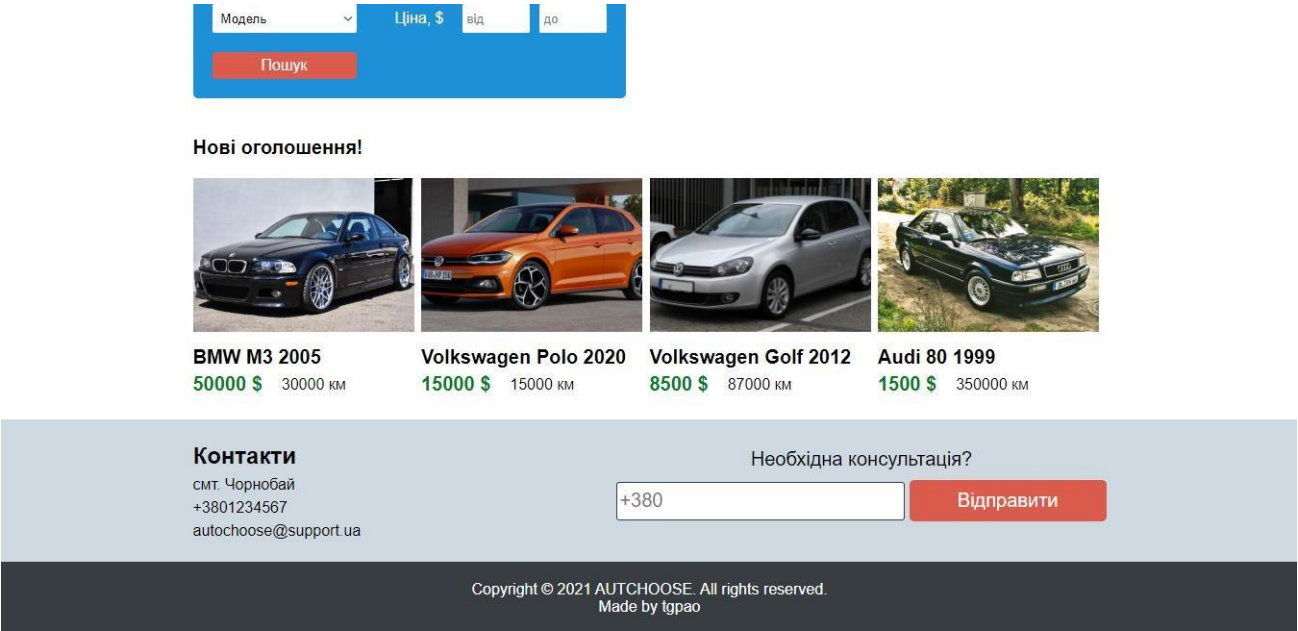
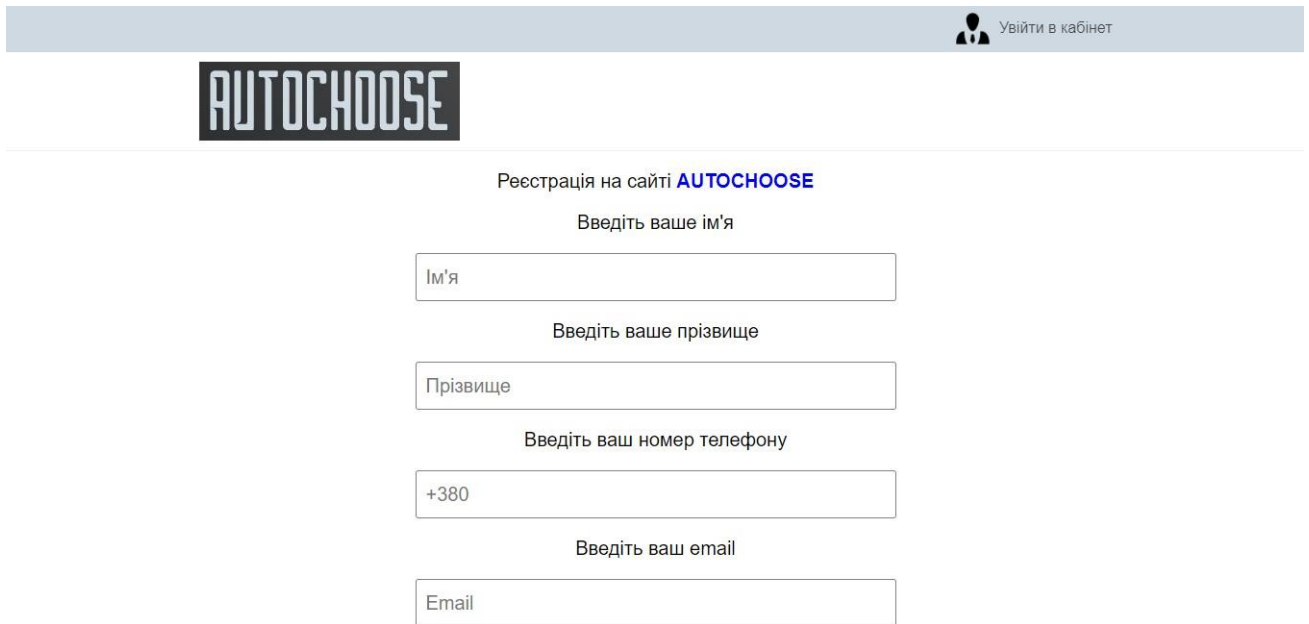


Рисунок 4.3 – Форма зворотного зв'язку

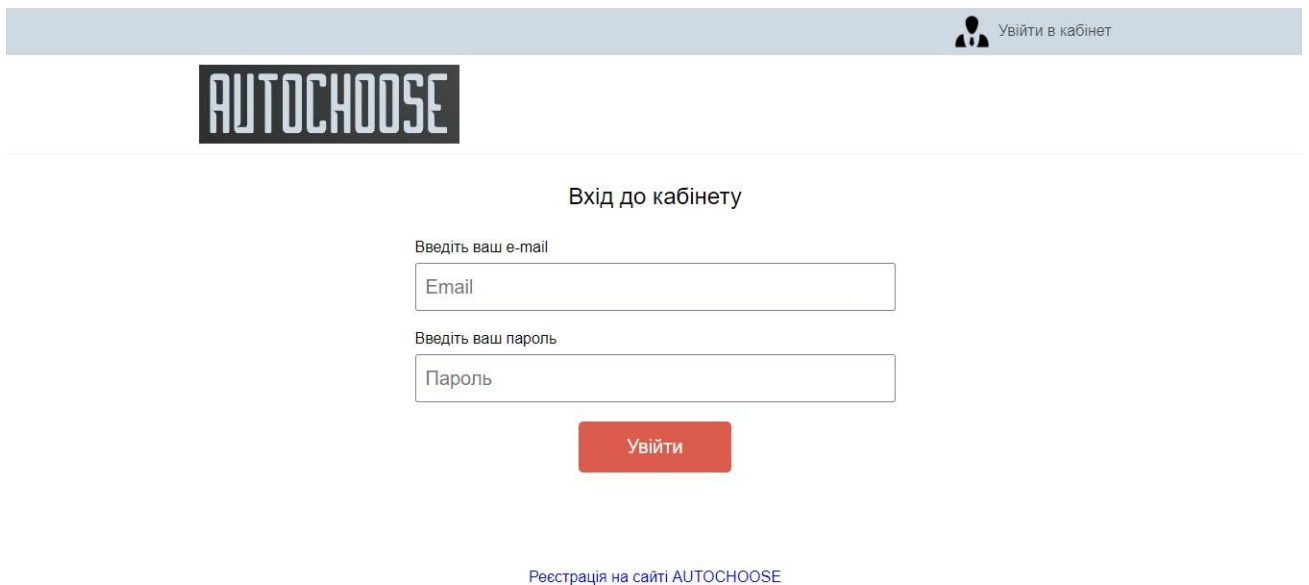
Користувач має змогу зареєструватися на сайті за допомогою форми реєстрації (рис. 4.4).



The screenshot shows the registration page of the AUTOCHOOSE website. At the top, there is a light blue header bar with a user icon and the text "Увійти в кабінет" (Login to cabinet). Below the header, the AUTOCHOOSE logo is displayed in a dark box. The main content area is titled "Реєстрація на сайті AUTOCHOOSE" (Registration on the AUTOCHOOSE website). It contains four input fields with labels: "Введіть ваше ім'я" (Enter your name), "Введіть ваше прізвище" (Enter your surname), "Введіть ваш номер телефону" (Enter your phone number), and "Введіть ваш email" (Enter your email). The phone number field has a pre-filled "+380".

Рисунок 4.4 – Сторінка реєстрації

Для авторизації необхідно перейти на сторінку авторизації (рис. 4.5).



The screenshot shows the login page of the AUTOCHOOSE website. At the top, there is a light blue header bar with a user icon and the text "Увійти в кабінет" (Login to cabinet). Below the header, the AUTOCHOOSE logo is displayed in a dark box. The main content area is titled "Вхід до кабінету" (Login to cabinet). It contains two input fields with labels: "Введіть ваш e-mail" (Enter your e-mail) and "Введіть ваш пароль" (Enter your password). Below the password field is a red button labeled "Увійти" (Login). At the bottom, there is a link "Реєстрація на сайті AUTOCHOOSE" (Registration on the AUTOCHOOSE website).

Рисунок 4.5 – Сторінка авторизації

Зареєстровані та авторизовані користувачі мають змогу створити власне оголошення за допомогою форми створення нових оголошень (рис. 4.6).

AUTOCHOOSE Продати авто

Додавання оголошення

Вкажіть характеристики вашого авто

Марка	Марка
Модель	Модель
Тип кузова	Тип кузова
Тип палива	Тип палива
Привід	Привід
КПП	КПП
Регіон	Регіон
Рік випуску	Рік випуску
Ціна, \$	Ціна
Пробіг, км	Пробіг
Опис	Опис

Розмістити оголошення

Рисунок 4.6 – Сторінка створення нового оголошення

Розглянемо детальніше розгортання програмного продукту.

Для роботи на сайті «AUTOCHOOSE» користувачеві потрібно виконати такі кроки:

1. Підключитися до мережі Інтернет.
2. Запустити браузер.
3. Ввести URL-адресу сайту.
4. Дочекатися відкриття сайту.

Коректна робота сайту буде забезпечена лише при стабільному підключенні до Інтернет-мережі.

Для роботи програмного продукту необхідно таке апаратне та програмне забезпечення і комп'ютерне приладдя:

1. Вимоги до апаратного забезпечення:
 - Об'єм RAM: > 512 Мб.
 - Відео пам'ять: > 128 Мб.
 - Процесор: > 1 ГГц.
2. Вимоги до програмного забезпечення:

- будь-яка сучасна операційна система;
- браузер, який підтримує сучасні технології WEB-програмування.

3. Комп'ютерне приладдя:

- Клавіатура;

Маніпулятор миша.

ВИСНОВКИ

Мета кваліфікаційної роботи – підвищення ефективності, зручності й надійності пошуку та підбору транспортного засобу за заданими критеріями, шляхом розробки веб-сайту «AUTOCHOOSE», досягнута.

Для реалізації поставленої задачі в роботі вирішені такі завдання:

- проведено порівняльний аналіз існуючих порталів пошуку транспортного засобу;
- здійснено вибір програмно-технічних засобів для реалізації завдань дослідження;
- спроектовано інформаційну систему з пошуку та підбору транспортного засобу у вигляді сайту «AUTOCHOOSE»;
- розроблено базу даних сайту «AUTOCHOOSE»;
- розроблено сайт з пошуку та підбору транспортного засобу «AUTOCHOOSE»;
- проведено тестування сайту «AUTOCHOOSE».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сайт «auto.ria.com». URL: <https://auto.ria.com/uk/> (дата звернення: 05.05.2023).
2. Сайт «rst.ua». URL: <https://rst.ua/ukr/> (дата звернення: 05.05.2023).
3. Сайт «uavto.com.ua». URL: <http://uavto.com.ua> (дата звернення: 05.05.2023).
4. Сайт «ab.ua». URL: <https://ab.ua> (дата звернення: 05.05.2023).
5. Сайт «autoportal.ua». URL: <https://autoportal.ua> (дата звернення: 05.05.2023).
6. Мова програмування JavaScript URL: <http://www.znannya.org/?view=javascript>
7. Кожем'якін А. А. HTML та CSS у прикладах. М. : Альтекс-А, 2014. 416 с.
8. SQL – мова БД SQL: її переваги, недоліки, оператори. URL: <http://programmersworld.xyz/article/18/62>
9. Ajax Systems. URL: https://uk.wikipedia.org/wiki/Ajax_Systems
10. Open Server Panel. URL: <https://ospanel.io>
11. Проектування бази даних. URL: https://old.pkng.pl.ua/index.php?option=com_content&view=article&id=192:2011-01-14-08-44-15&catid=19:2011-01-14-08-34-01&Itemid=7
12. Інформаційна система. URL: <http://www.kievoit.ippo.kubg.edu.ua/kievoit/2013/95/95.html>
13. Бізнес правила. URL: https://studme.org/138831/ekonomika/vidy_biznes_pravil

ДОДАТОК А**ІНФОРМАЦІЙНА СИСТЕМА З ПОШУКУ ТА ПІДБОРУ ТРАНСПОРТНОГО
ЗАСОБУ**

Текст програми

Листів 9

Розробник

Ломака М. А.

В додатку наведено програмний код головних модулів та сторінок інформаційно-консультаційної системи з пошуку та підбору транспортного засобу (фрагмент).

Фрагмент коду головної сторінки:

```
<?php include 'connection.php'; ?>
<!DOCTYPE html>
<html>
<head>
<title>AutoChoose</title>
<meta charset="utf-8">
<link rel="stylesheet" href="css/style.css">
<link rel="shortcut icon" type="image/x-icon" href="img/favicon.png">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.min.js"></script>
</head>
<body>
<?php
include 'header.php';
$brand_queried = mysqli_query($connection, "SELECT * FROM `brand` ORDER BY `brand`");
$recent_queried = mysqli_query($connection, "SELECT * FROM `announcement`
INNER JOIN brand ON announcement.brand_id = brand.id_brand
INNER JOIN model ON announcement.model_id = model.id_model
ORDER by id_announcement DESC LIMIT 4");
?>
<div class="search">
<div class="container">
<div class="search__inner">
<form method="POST" action="announcements.php" class="search__form">
<div class="search__input">
<div class="search__main">
<div class="search__model">
<select id="brand" name="brand" class="input__field">
<option disabled selected>Марка</option>
<?php
while ($brand = mysqli_fetch_assoc($brand_queried)) {
echo '<option value="'. $brand[brand].'">' . $brand[brand] . '</option>';
}
?>
</select>
</div>
<div class="search__model">
<select id="model" name="model" class="input__field">
<option disabled selected>Модель</option>
</select>
</div>
</div>
<div class="search__secondary">
<div class="search__region">
<div class="search__text">Регіон</div>
<div>
<input class="input__field" type="text" name="region" placeholder="Регіон">
</div>
</div>
<div class="search__region">
<div class="search__text">Ціна, $</div>
<div class="price__fields">
```

```

<input class="input__field price" type="text" name="priceFrom" placeholder="Від">
<input class="input__field price" type="text" name="priceTo" placeholder="до">
</div>
</div>
</div>
</div><!-- search__input -->
<div class="search__send">
<input class="submit__btn" type="submit" name="send" value="Пошук">
</div><!-- search__send -->
</form>
</div><!-- search__inner -->
</div><!-- container -->
</div><!-- search -->
<div class="recent">
<div class="container">
<div class="recent__new">Нові оголошення!</div>
<div class="recent__inner">
<?php
while($recent_assoced = mysqli_fetch_assoc($recent_queried)){
$select_img_query = mysqli_query($connection, "SELECT * FROM uploaded_img WHERE
announcement_id = '$recent_assoced[id_announcement]'");
$select_img_assoced = mysqli_fetch_assoc($select_img_query);
?>
<div class="recent__item">
<div class="recent__img"></div>
<form method="POST" action="selectedcar.php">
<button class="announcement__btn" name="selectedcar" value="<?php echo
$recent_assoced[id_announcement]; ?>">
<div class="recent__title"><?php echo $recent_assoced[brand] . ' '. $recent_assoced[model] . ' ' .
$recent_assoced[year]; ?>
</div>
</button>
</form>
<div class="recent__text">
<div class="recent__price"><?php echo $recent_assoced[price]; ?> $</div>
<div class="recent__mileage"> <?php echo $recent_assoced[mileage]; ?> км</div>
</div>
</div><!-- recent__item -->
<?php } ?>
</div><!-- recent__inner -->
</div><!-- container -->
</div><!-- recent -->
<?php
include 'footer.php';
?>
<script src="ajax.js"></script>
</body>
</html>

```

Фрагмент коду сторінки оголошень:

```

<?php include 'connection.php'; ?>
<!DOCTYPE html>
<html>
<head>

```

```

<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="css/style.css">
<link rel="shortcut icon" type="image/x-icon" href="img/favicon.png">
<title>
<?php
if(isset($_POST[brand])){
echo $_POST[brand]. ' '.$_POST[model];;
} else {
echo 'ABTO';
}
?>
</title>
</head>
<body>
<?php
include 'header.php';
$select_announcement = "SELECT * FROM announcement
INNER JOIN brand ON announcement.brand_id = brand.id_brand
INNER JOIN model ON announcement.model_id = model.id_model
INNER JOIN body ON announcement.body_id = body.id_body
INNER JOIN fuel ON announcement.fuel_id = fuel.id_fuel
INNER JOIN drive_type ON announcement.drive_type_id = drive_type.id_drive_type
INNER JOIN gearbox ON announcement.gearbox_id = gearbox.id_gearbox
INNER JOIN users ON announcement.users_id = users.id_users";
$body_queried = mysqli_query($connection, "SELECT * FROM body");
$fuel_queried = mysqli_query($connection, "SELECT * FROM fuel");
$drive_type_queried = mysqli_query($connection, "SELECT * FROM drive_type");
$gearbox_queried = mysqli_query($connection, "SELECT * FROM gearbox");
if (isset($_POST[brand]) && empty($_POST[model])) {
$select_announcement_query = mysqli_query($connection, "$select_announcement WHERE brand =
'$_POST[brand]'");
}
if (isset($_POST[model])) {
$select_announcement_query = mysqli_query($connection, "$select_announcement WHERE model
= '$_POST[model]'");
}
if (empty($_POST[brand])) {
$select_announcement_query = mysqli_query($connection, $select_announcement);
}
?>

<div class="announcement">
<div class="container">
<div class="announcement__inner">
<div class="announcement__cars">
<?php
while ($select_announcement_assoced = mysqli_fetch_assoc($select_announcement_query)) {
$select_img_query = mysqli_query($connection, "SELECT * FROM uploaded_img WHERE
announcement_id = '$select_announcement_assoced[id_announcement]'");
$select_img_assoced = mysqli_fetch_assoc($select_img_query);
?>
<div class="announcement__item">
<div class="announcement__photo"></div>
<div class="announcement__info">

```

```

<form method="POST" action="selectedcar.php">
<button      class="announcement__btn"      name="selectedcar"      value="<?php      echo
$select_announcement_assoc[id_announcement]; ?>">
<div class="announcement__title"><?php echo $select_announcement_assoc[brand] . ' ' .
$select_announcement_assoc[model] . ' ' . $select_announcement_assoc[year]; ?>
</div>
</button>
</form>
<div class="announcement__price"><?php echo $select_announcement_assoc[price]?> $</div>
<div class="announcement__data">
<div class="data__left">
<div class="data__text"><?php echo $select_announcement_assoc[mileage]?> км</div>
<div class="data__text"><?php echo $select_announcement_assoc[fuel]?></div>
</div>
<div class="data__right">
<div class="data__text"><?php echo $select_announcement_assoc[region]?></div>
<div class="data__text"><?php echo $select_announcement_assoc[gearbox]?></div>
</div>
</div>
<div      class="announcement__description"><?php      echo
$select_announcement_assoc[description]?></div>
</div>
</div>
<?php } ?>
</div><!-- announcement__cars -->
<div class="announcement__filter">
<p class="announcement__filter__title">Фільтр авто</p>
<select id="body" name="body" class="filter__input">
<option disabled selected>Кυзов</option>
<?php
while ($body = mysqli_fetch_assoc($body_queried)) {
echo '<option value="'. $body[id_body].'">' . $body[body] . '</option>';
}
?>
</select>
<select id="fuel" name="fuel" class="filter__input">
<option disabled selected>Тип палива</option>
<?php
while ($fuel = mysqli_fetch_assoc($fuel_queried)) {
echo '<option value="'. $fuel[id_fuel].'">' . $fuel[fuel] . '</option>';
}
?>
</select>
<select id="drive_type" name="drive_type" class="filter__input">
<option disabled selected>Привід</option>
<?php
while ($drive_type = mysqli_fetch_assoc($drive_type_queried)) {
echo '<option value="'. $drive_type[id_drive_type].'">' . $drive_type[drive_type] . '</option>';
}
?>
</select>
<select id="gearbox" name="gearbox" class="filter__input">
<option disabled selected>КПП</option>
<?php
while ($gearbox = mysqli_fetch_assoc($gearbox_queried)) {
echo '<option value="'. $gearbox[id_gearbox].'">' . $gearbox[gearbox] . '</option>';
}
?>
</select>

```

```

}
?>
</select>
<input type="text" name="year" class="filter__input input__not__select" placeholder="Рік">
<input type="text" name="region" class="filter__input input__not__select" placeholder="Регіон">
</div><!-- announcement__filter -->
</div><!-- announcement__inner -->
</div><!-- container -->
</div><!-- announcement -->
<?php
include 'footer.php';
?>
</body>
</html>

```

Фрагмент коду форми створення нового оголошення:

```

<?php include 'connection.php'; ?>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="css/style.css">
<link rel="shortcut icon" type="image/x-icon" href="img/favicon.png">
<title>Продати авто</title>
</head>
<body>
<?php
include 'header.php';
?>
<div class="container">
<div class="sellcar">
<div class="sellcar__title">Додавання оголошення</div>
<div>Вкажіть характеристики вашого авто</div>
<form class="sellcar__form">
<div class="sellcar__info">
<div class="sellcar__info__text">Марка</div>
<input type="text" name="brand" placeholder="Марка" class="sellcar__info__input">
</div>
<div class="sellcar__info">
<div class="sellcar__info__text">Модель</div>
<input type="text" name="brand" placeholder="Модель" class="sellcar__info__input">
</div>
<div class="sellcar__info">
<div class="sellcar__info__text">Тип кузова</div>
<input type="text" name="brand" placeholder="Тип кузова" class="sellcar__info__input">
</div>
<div class="sellcar__info">
<div class="sellcar__info__text">Тип палива</div>
<input type="text" name="brand" placeholder="Тип палива" class="sellcar__info__input">
</div>
<div class="sellcar__info">
<div class="sellcar__info__text">Привід</div>
<input type="text" name="brand" placeholder="Привід" class="sellcar__info__input">
</div>

```

```

<div class="sellcar__info">
<div class="sellcar__info__text">КПП</div>
<input type="text" name="brand" placeholder="КПП" class="sellcar__info__input">
</div>
<div class="sellcar__info">
<div class="sellcar__info__text">Регіон</div>
<input type="text" name="brand" placeholder="Регіон" class="sellcar__info__input">
</div>
<div class="sellcar__info">
<div class="sellcar__info__text">Рік випуску</div>
<input type="text" name="brand" placeholder="Рік випуску" class="sellcar__info__input">
</div>
<div class="sellcar__info">
<div class="sellcar__info__text">Ціна, $</div>
<input type="text" name="brand" placeholder="Ціна" class="sellcar__info__input">
</div>
<div class="sellcar__info">
<div class="sellcar__info__text">Пробіг, км</div>
<input type="text" name="brand" placeholder="Пробіг" class="sellcar__info__input">
</div>
<div class="sellcar__info">
<div class="sellcar__info__text">Опис</div>
<input type="text" name="brand" placeholder="Опис" class="sellcar__info__input">
</div>
<div class="sellcar__info__submit">
<input type="submit" name="do_sellcar" value="Розмістити оголошення" class="btn">
</div>
</form>
</div><!-- sellcar -->
</div><!-- container -->
<?php
include 'footer.php';
?>
</body>
</html>

```

Фрагмент коду сторінки реєстрації:

```

<?php include 'connection.php'; ?>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="css/style.css">
<link rel="shortcut icon" type="image/x-icon" href="img/favicon.png">
<title>Реєстрація</title>
</head>
<body>
<?php
include 'header.php';
$name = $_POST['name'];
$surname = $_POST['surname'];
$number = $_POST['phonenummer'];
$email = $_POST['email'];
$password = $_POST['password'];

```

```

$password_check = $_POST['password_check'];
$do_register = $_POST['do_register'];
$password_hashed = password_hash($password, PASSWORD_DEFAULT);
$insert = "INSERT INTO `users` (`id_users`, `name`, `surname`, `email`, `phone_number`,
`password`) VALUES (NULL, '$name', '$surname', '$email', '$number', '$password_hashed')";
$countemail = mysqli_query($connection, "SELECT email FROM `users` WHERE `email` =
'$email'");
$countnumber = mysqli_query($connection, "SELECT phone_number FROM `users` WHERE
`phone_number` = '$number'");
if (isset($do_register)) {
$errors = array();
if (trim($name) == "") {
$errors[] = "Введіть ваше ім'я!";
}
if (trim($surname) == "") {
$errors[] = "Введіть ваше прізвище!";
}
if (trim($number) == "") {
$errors[] = "Введіть ваш номер телефону!";
}
if (iconv_strlen($number) != '13') {
$errors[] = "Номер введено невірно!";
}
if (mysqli_num_rows($countnumber) > 0) {
$errors[] = "Такий номер телефону вже зареєстрований!";
}
if (trim($email) == "") {
$errors[] = "Введіть ваш email!";
}
if (mysqli_num_rows($countemail) > 0) {
$errors[] = "Така email адреса вже зареєстрована!";
}
if (trim($password) == "") {
$errors[] = "Введіть ваш пароль!";
}
if ($password_check != $password) {
$errors[] = "Паролі не співпадають!";
}
if (empty($errors)) {
mysqli_query($connection, $insert);
echo '<script>
location.href = "login.php"
</script>';
} else { ?>
<div class="container">
<div class="reg__errors">
<?php echo array_shift($errors); ?>
</div>
</div>
<?php }
}
?>
<div class="registration">
<div class="container">
<form action="registration.php" method="post" class="registration__form">
<div class="reg__site">

```

```

<div class="reg_site_text">Регістрація на сайті <strong>
class="reg_site_name">AUTOCHOOSE</strong></div>
</div>
<p class="registration__text">Введіть ваше ім'я</p>
<input type="text" name="name" placeholder="Ім'я" class="registration__input" value="<?php echo $name; ?>">
<p class="registration__text">Введіть ваше прізвище</p>
<input type="text" name="surname" placeholder="Прізвище" class="registration__input" value="<?php echo $surname; ?>">
<p class="registration__text">Введіть ваш номер телефону</p>
<input type="text" name="phonenumber" placeholder="+380" class="registration__input" value="<?php echo $number; ?>">
<p class="registration__text">Введіть ваш email</p>
<input type="text" name="email" placeholder="Email" class="registration__input" value="<?php echo $email; ?>">
<p class="registration__text">Введіть ваш пароль</p>
<input type="password" name="password" placeholder="Пароль" class="registration__input" value="<?php echo $password; ?>">
<p class="registration__text">Введіть ваш пароль ще раз</p>
<input type="password" name="password_check" placeholder="Перевірка паролю" class="registration__input" value="<?php echo $password_check; ?>">
<br><input type="submit" name="do_register" class="btn reg" value="Продовжити">
</form>
</div><!-- container -->
</div><!-- registration -->
<?php
include 'footer.php';
?>
</body>
</html>

```

Фрагмент коду сторінки авторизації:

```

<?php
include 'connection.php';
?>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="css/style.css">
<link rel="shortcut icon" type="image/x-icon" href="img/favicon.png">
<title>Увійти в кабінет</title>
</head>
<body>
<?php
include 'header.php';
$email = $_POST['email'];
$password = $_POST['password'];
$do_login = $_POST['do_login'];
if (isset($do_login)) {
$errors = array();
$findemail = mysqli_query($connection, "SELECT email FROM `users` WHERE `email` = '$email'");
$email_assoced = mysqli_fetch_assoc($findemail);

```

```

$findpassword = mysqli_query($connection, "SELECT password FROM `users` WHERE `email` = '$email'");
$password_assoc = mysqli_fetch_assoc($findpassword);
$passwordhashed = $password_assoc['password'];
if (mysqli_num_rows($findemail) > 0) {
if (password_verify($password, $passwordhashed)) {
$user_queried = mysqli_query($connection, "SELECT * FROM `users` WHERE `email` = '$email'");
$user_assoc = mysqli_fetch_assoc($user_queried);
$_SESSION['logged_user'] = $user_assoc;
echo '<script>
location.href = "index.php"
</script>';
} else {
$errors[] = 'The password is wrong';
}
} else {
$errors[] = 'The email is wrong';
}
if (! empty($errors)) { ?>
<div class="login__error">
<?php echo array_shift($errors); ?>
</div>
<?php }
}
?>
<div class="login">
<div class="container">
<form class="login__form" action="login.php" method="post">
<p class="login__text">Вхід до кабінету</p>
<div>
<div class="input__text">Введіть ваш e-mail</div>
<input class="login__input" type="text" name="email" placeholder="Email" value="<?php echo $email; ?>">
</div>
<div>
<div class="input__text">Введіть ваш пароль</div>
<input class="login__input" type="password" name="password" placeholder="Пароль" value="<?php echo $password; ?>">
</div>
<div>
<input class="btn" type="submit" name="do_login" value="Увійти">
</div>
</form>
<div class="reg__btn">
<a href="registration.php" class="reg__link">Реєстрація на сайті AUTOCHOOSE</a>
</div>
</div><!-- container -->
</div><!-- login -->
<?php
include 'footer.php';
?>
</body>
</html>

```