

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування інституту, факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ ПОШУКУ АВТОМОБІЛІВ У
ПОЗШУКУ. WEB APPLICATION DEVELOPMENT FOR WANTED CAR
SEARCH

Виконав: студент 4 курсу, групи КНТ-129сп
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

Недошивко В.С.

(прізвище та ініціали)

Керівник Казурова А.Є.

(прізвище та ініціали)

Рецензент Голуб Т.В.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 121 Інженерія програмного забезпечення
 (код і найменування)
 Освітня програма (спеціалізація) Інженерія програмного забезпечення
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 2022 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Недошивка Віталія Сергійовича
 (прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка вебзастосунку для пошуку автомобілів у розшуку. Web Application Development for Wanted Car Search

керівник проєкту (роботи) Казурова Аліна Євгенівна, к.т.н., доцент,
 (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “ 27 ” квітня 2022 року № 102

2. Строк подання студентом проєкту (роботи) 07 червня 2022 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Опис програми. 3. Розробка програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	Казурова А.Є., доцент		
Нормоконтроль	Дейнега Л.Ю., ст. викладач		

7. Дата видачі завдання « 04 » квітня 2022 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій	2 тиждень	Розділ 2
	розробки.		
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділ 3, 4
6	Тестування та експериментальне дослідження	5 тиждень	Розділ 4
	програмного забезпечення.		
7	Оформлення пояснювальної записки та документів	6 тиждень	Додатки
	до неї.		
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис)

Недошивко В.С.

(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис)

Казурова А.Є.

(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 64 с., 13 рис., 4 табл., 3 дод., 20 джерел.

ВЕБСАЙТ, АВТОМОБІЛЬ, ПОШУК, КЛІЄНТ, СЕРВЕР, CSS, HTML, JS, REACT.

Об'єкт дослідження – програмні засоби для пошуку транспортних засобів у розшуку.

Предмет дослідження – вебсайт для пошуку транспортних засобів у розшуку.

Мета роботи – створення вебзастосунок для швидкого пошуку транспортних засобів у розшуку.

Матеріали, методи та технічні засоби: функціональне та об'єктно-орієнтоване програмування, мови програмування JavaScript, персональний комп'ютер з процесором Intel Core 2 Duo під управлінням операційної системи Microsoft Windows 10.

Результати. Створено вебзастосунок який допомагає знайти транспортні засоби у розшуку.

Висновки. Розроблено вебзастосунок для пошуку транспортних засобів у розшуку, який автоматизує отримання та відображення даних з сайту відкритих даних.

Галузь використання – допомога в пошуку транспортних засобів у розшуку.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 64 pages, 13 figures, 4 tables, 3 appendixes, 20 sources.

WEBSITE, VEHICLE, SEARCH, CLIENT, SERVER, CSS, HTML, JS, REACT.

The object of research is software for searching for wanted vehicles.

The subject of the research is a website for searching for wanted vehicles.

The purpose of the work is to create a web application for quick search of wanted vehicles.

Materials, methods and technical means: functional and object-oriented programming, JavaScript programming languages, personal computer with Intel Core 2 Duo processor running Microsoft Windows 10 operating system.

Results. A web application has been created to help find wanted vehicles.

Conclusions. Developed a web application for searching for vehicles in search, which automates the receipt and display of data from the site of open data.

Field of use – assistance in finding wanted vehicles.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз предметної області.....	10
1.1 Основні поняття	10
1.2 Опис предметної області	10
1.3 Огляд функціональних можливостей існуючих систем	12
1.3.1 Вебсайт УКРНОМЕР	12
1.3.2 Вебсайт АВТОНОМЕР.....	16
1.4 Визначення вимог до інформаційної системи	17
1.5 Висновок	19
2 Опис програми.....	20
2.1 Технології створення web-додатків.....	20
2.2 Вибір мови програмування для серверної частини	22
2.3 Обґрунтування вибору клієнтської частини	24
2.4 Аналіз вибору СКБД.....	28
2.5 Висновок	33
3 Розробка програми	34
3.1 Загальний опис системи	34
3.2 Архітектура системи.....	34
3.3 Висновок	36
4 Експлуатація, тестування та експериментальне дослідження програми	37
4.1 Опис застосування програми	37
4.2 Керівництво програміста.....	37
4.2.1 Звернення до програмного продукту (файли проекту)	38
4.3 Керівництво користувача.....	38
4.4 Висновок	40
Висновки	41
Перелік джерел посилання	42

Додаток А Технічне завдання	45
Додаток Б Текст файлів програми.....	49
Додаток В Слайди презентації.....	58

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API	– Application Programming Interface;
CSS	– Cascading Style Sheets;
DOM	– Document Object Model;
GPS	– Global Positioning System;
HTML	– HyperText Markup Language;
IDE	– Integrated Development Environment;
MIT	– Massachusetts Institute of Technology;
REST	– Representational State Transfer;
SSR	– Server Side Rendering;
VIN	– Vehicle identification number;
ДАІ	– державна автомобільна інспекція;
ПП	– програмний продукт;
СКБД	– система керування базами даних.

ВСТУП

З кожним роком кількість вкрадених автомобілів збільшується. Пошук таких автівок є доволі складною роботою. Їх можна знайти за допомогою GPS-трекера, якщо такі встановлені, але їх треба встановлювати як додаткове обладнання. Цей веб застосунок повинен допомогти знайти вкрадений автомобіль. Нажаль самотійно він не може працювати, але він може працювати з допомогою людей, які хочуть допомогти знайти автомобіль. Коли ця людина знайде підозрілий автомобіль, який довго стоїть та має закислі гальма, пил на автомобілі або зламану личинку замка двері, повинна перейти на сторінку пошуку в вебзастосунку, знайти автомобіль за реєстраційним або VIN-номером, а потім зателефонувати поліції.

Зазвичай після викрадення автівки її ставлять у місце на місяць-два, щоб побачити чи немає за неї стеження, а потім її розбирають на деталі, або перепродають. Тому цей вебзастосунок може допомогти знайти автомобілі в період відстоювання автівки.

Вебзастосунок розроблений за допомогою сучасних технологій, що дозволяє йому відповідати на запити пошуку максимально швидко.

Цей вебзастосунок та небайдужі люди можуть допомогти знайти більше вкрадених автомобілів та повернути їх власникам.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Основні поняття

Вебсайт – одна або кілька логічно пов'язаних між собою вебсторінок; також місце розташування контенту сервера. Зазвичай сайт в інтернеті являє собою масив пов'язаних даних, що мають унікальну адресу і сприймається користувачами як єдине ціле. Вебсайти називаються так, тому що доступ до них відбувається по протоколу HTTP [1].

Клієнт – це апаратний або програмний компонент обчислювальної системи, який посилає запити до сервера.

Сервер – це апаратний або програмний компонент обчислювальної системи, який приймає запити від клієнта та обчислює їх.

Транспортний засіб – це технічний пристрій, призначений для перевезення людей і вантажів [2].

1.2. Опис предметної області

Кожного дня на вулицях України викрадають декілька автомобілів. За минулий рік викрали 1475 автомобілів, звісно, зрівнюючи с минулими роками, відразу видно позитивна тенденція в сторону автомобілістів, але покладатися на неї не варто. Наприклад, технології зловмисників стрімко розвиваються, і неможливо передбачити тенденції 2022.

Топ марок, що викрадаються, в Україні (2021 р.):

- Тойота;
- Лексус;
- Кіа;
- Мазда;
- Хонда;
- БМВ;
- Ленд Ровер;

- Порш;
- Ауді [3].

Власникам цих марок потрібно буди більш обачними до своїх автомобілів, та встановлювати прилади для захисту від викрадення.

Частіше всього зловмисники використовують кодграбер для викрадання авто. Кодграбер – пристрій, що перехоплює сигнал від брелока, яким власник ставить свій автомобіль в охорону [3].

Для протидії викраденням у вересні 2020 року було прийнято Закон «Про внесення змін до статті 289 Кримінального кодексу України щодо протидії незаконному заволодінню транспортним засобом». І відтепер викрадач несе кримінальну відповідальність за крадіжку авто та йому загрожує позбавлення волі на термін від 3-х до 5-ти років. Незважаючи на це, злочини продовжують викрадати автомобілі [4].

Після викрадення автомобіль завжди, йде у місце відстійника, де знаходиться не менше двох тижнів, а частіше довше. Це викликано необхідністю викрадача забезпечити собі безпеку. Державний знак під час викрадення змінюється на вкрадений, а автомобіль переміщається у сусідній район чи квартал, де перебуває деякий час. Причому ці машини дуже легко виділити з більшості. Як правило, вони виглядають занедбаними та безхазяйними, а при детальній перевірці державного реєстраційного знака виявляється, що номер встановлений від іншого автомобіля і це є причиною бити на сполох і шукати справжнього власника [5].

Вебзастосунок допоможе людині перевірити автомобіль за державним номером, та зрозуміти чи вкрадений автомобіль.

Для пошуку програмний продукт використовує відкриті дані. Відкриті дані – це концепція, за якою певні дані мають бути вільними для використання та розповсюдження будь-якою особою, за умов дотримання правил атрибуції та/або share-alike ліцензії [6].

У 2015 році був прийнятий Закон № 319 та Постанова Кабінету Міністрів України № 835. На підставі цих двох документів державні органи

зобов'язані надавати публічну інформацію в формі відкритих даних і регулярно оновлювати її на єдиному державному вебпорталі відкритих даних в певних форматах [7].

1.3 Огляд функціональних можливостей існуючих систем

На даний час вебсайти для пошуку дуже поширені, ми розглянемо деякі аналогічні системи, оцінимо їх переваги та недоліки.

1.3.1 Вебсайт УКРНОМЕР

Першим був розглянуто вебсайт «УКРНОМЕР», який знаходиться за посиланням <https://ukrnomer.com/ugon>. Головна сторінка сайту цього закладу представлена на рис. 1.1.

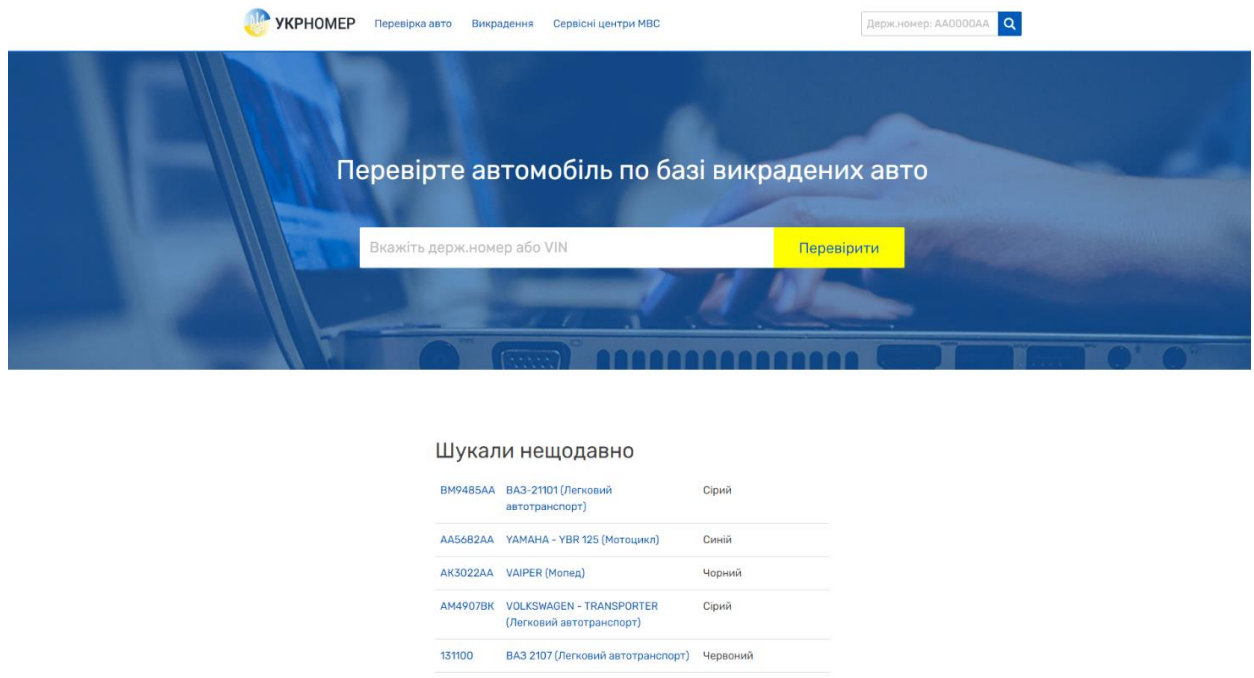


Рисунок 1.1 – Головна сторінка вебсайту «УКРНОМЕР»

До переваг варто віднести:

- простий та зрозумілий інтерфейс;

- працює швидко;
- має сторінку інформації про сервісні центри МВС;
- має розділ «Шукали нещодавно», який дозволяє повторно знаходити транспортні засоби які шукали.

Якщо ми введемо номер та натиснемо перевірити, то нас перенесе на сторінку з інформацією про автомобіль у розшуку (рис. 1.2).

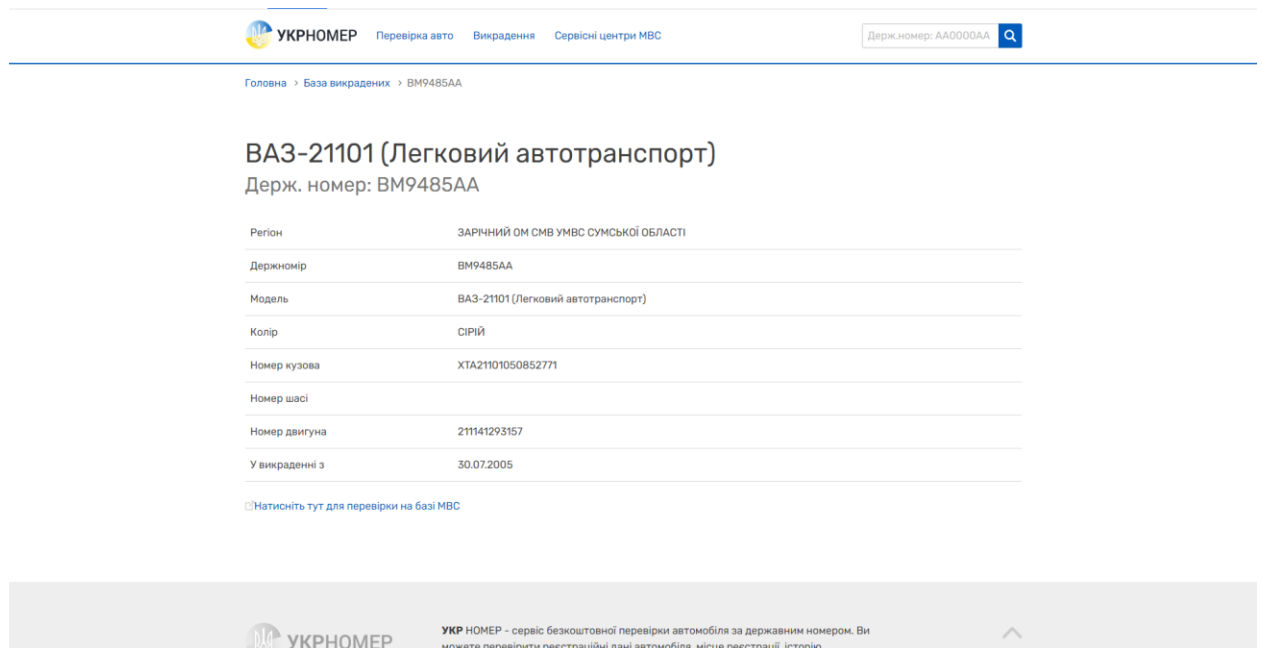


Рисунок 1.2 – Сторінка інформації про транспортний засіб у розшуку

Сторінка інформації про автомобіль відображає:

- марку та модель;
- регіон;
- державний реєстраційний номер;
- колір;
- номер кузова;
- номер шасі;
- номер двигуна;
- дату викрадення.

Якщо автомобіль не знайдено у базі угону, то буде відображена сторінка «Не знайдено» (рис. 1.3).

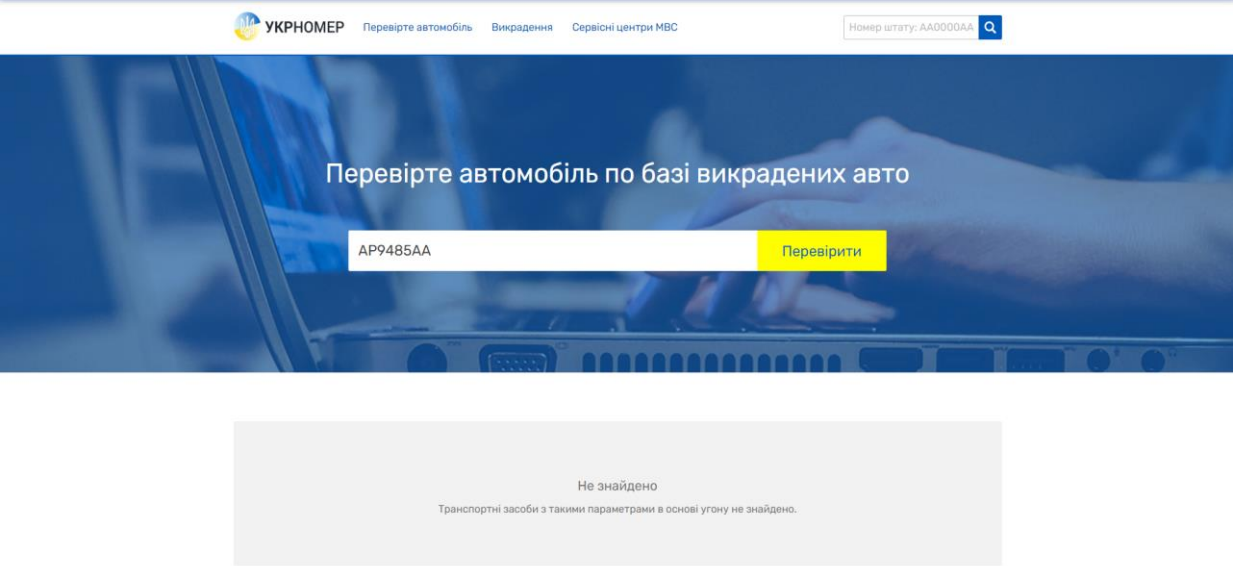


Рисунок 1.3 – Сторінка «Не знайдено» вебсайту «УКРНОМЕР»

Сторінка пошуку автомобілів за базою ДАІ на рис. 1.4. На сайті також є сторінка з перевірки автомобіля за базою ДАІ, на ній відображається основна інформація із бази ДАІ (рис. 1.5).

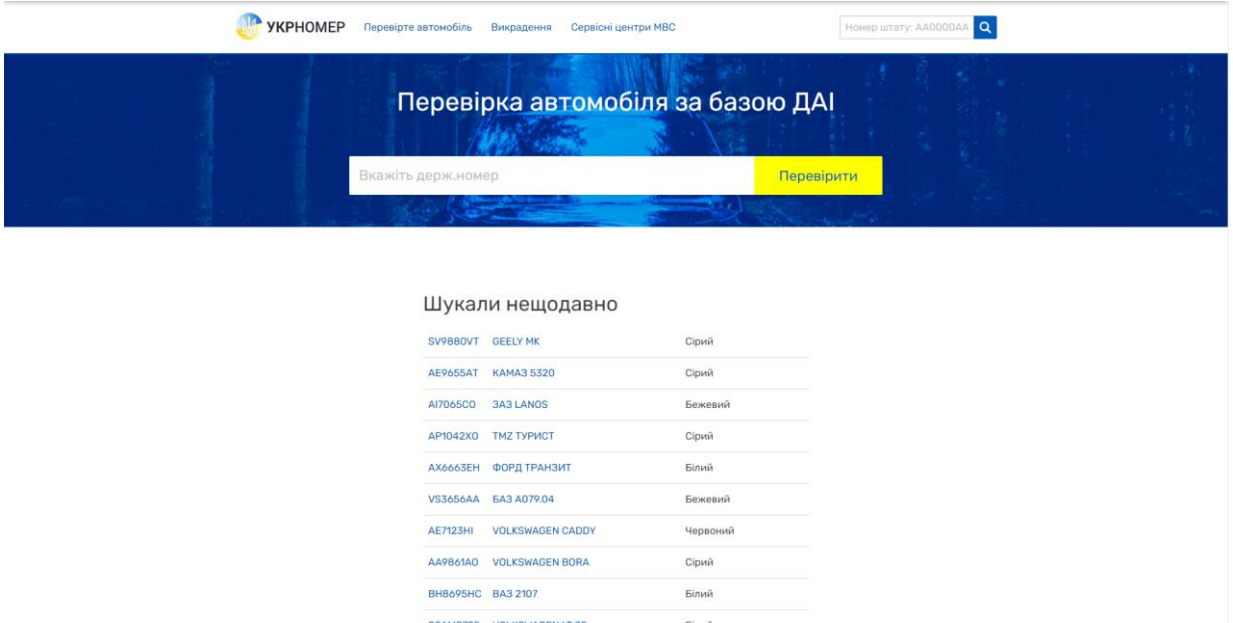


Рисунок 1.4 – Сторінка пошуку автомобілів за базою ДАІ

Також на сайті є можливість подивитись сервісні центри МВС та детальну інформацію про них (рис. 1.6, 1.7).

УКРНОМЕР

Перевірте автомобіль

Викрадення

Сервісні центри МВС

Номер штату: AA0000AA

Зведені дані

Перевірка на викрадення

Історія реєстраційних дій

Перевірка полісу ОСАГВ

Головна

Перевірте автомобіль

GEELY MK C89880BT

GEELY MK - 2009 р.в

Держ. номер: C89880BT

Место Регистрации: Чернігівська Область/М.Чернігів, Прилуки

Рік виробництва

2009 Рік

Об'єм двигуна

1587 см³

Тип

Легковий Седан-В

Паливо

Бензин Або Газ

Держ. номер

SV9880VT

Завантажити всі дані у PDF

Зведені дані

Власник	Приватна особа	Кузов	Седан-В
Бренд	Geely Mk	Призначення	Загальний
Модель	Mk	Паливо	Бензин Або Газ
Рік випуску	2009 рік	Об'єм двигуна	1587 см ³
Колір	Сірий	Споряджена маса	1070 кг.
Тип	Легковий	Повна маса	1460 кг.

Перевірка на викрадення

За нашими даними авто GEELY MK з держномером C89880BT не значиться у викраденні. Натисніть тут для перевірки на базі МВС.

Історія реєстраційних дій

№	Дата	Опис операції	відділ
1	06.05.2018	315 - перереєстрація тз на нов. власн. по договору укладеному в тсц	TSC 8046

Перевірка автентичності поліса ОСАГО

Натисніть тут для перевірки полісу страхування на базі МТСБУ

Рисунок 1.5 – Сторінка інформації про автомобіль за базою ДАІ

УКРНОМЕР

Перевірте автомобіль

Викрадення

Сервісні центри МВС

Номер штату: AA0000AA

Головна

Сервісні центри МВС

Сервісні центри МВС



АР Крим та м. Севастополь

Вінницька область

Волинська область

м. Київ

Дніпропетровська область

Донецька область

Житомирська область

Закарпатська область

Запорізька область

Івано-Франківська область

Київська область

Кіровоградська область

Луганська область

Львівська область

Миколаївська область

Одеська область

Полтавська область

Рівненська область

Сумська область

Тернопільська область

Харківська область

Херсонська область

Хмельницька область

Черкаська область

Чернігівська область

Чернівецька область

Рисунок 1.6 – Інформація про сервісні центри МВС

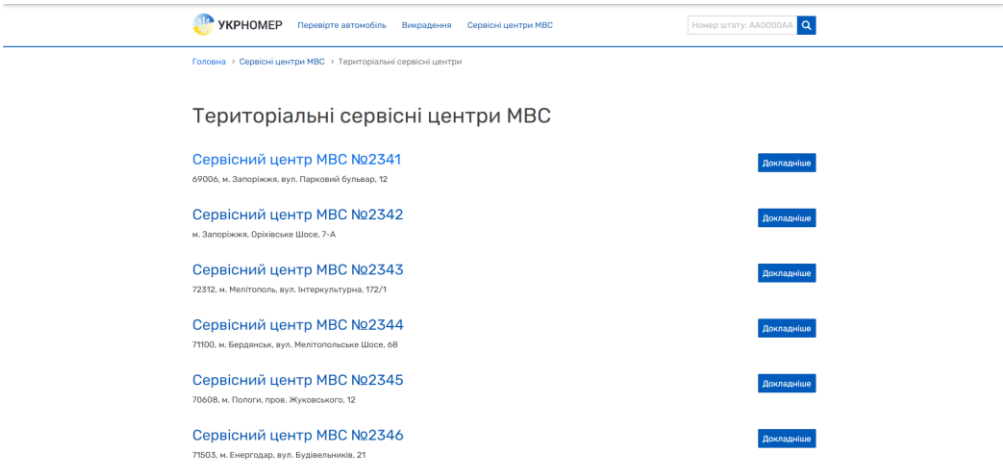


Рисунок 1.7 – Детальна інформація про сервісні центри МВС

1.3.2 Вебсайт АВТОНОМЕР

Наступним був розглянуто вебсайт «АВТОНОМЕР», він знаходиться за посиланням <https://ua.carplates.app>. Головна сторінка сайту цього закладу представлена на рис. 1.8.

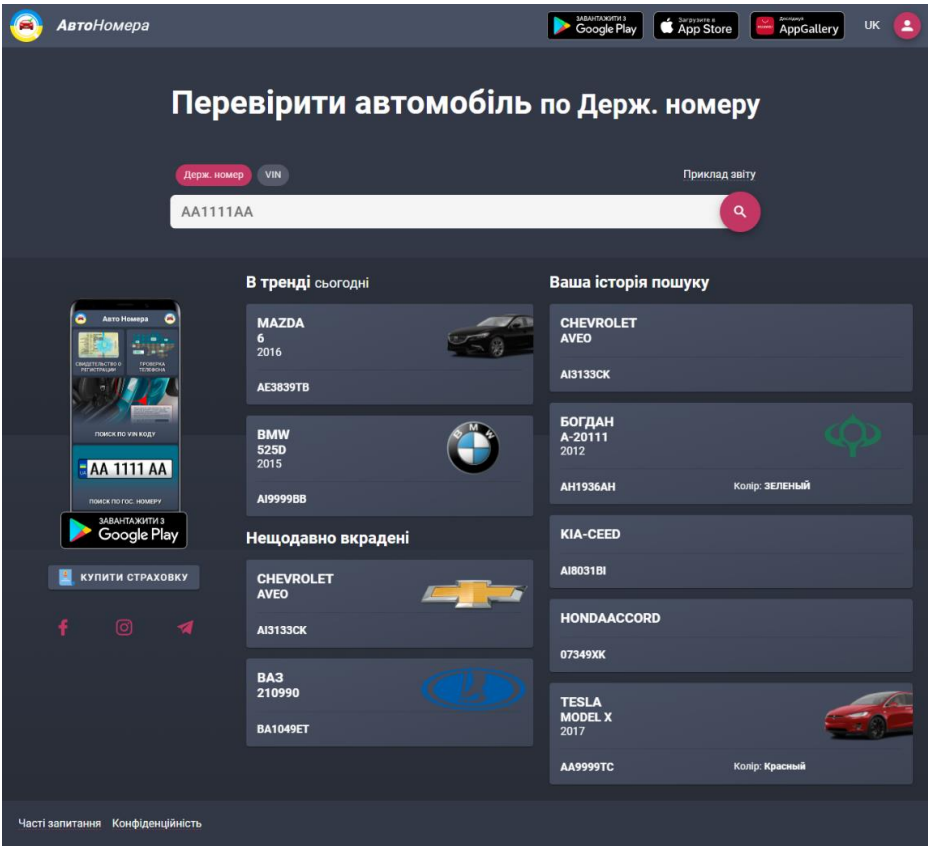


Рисунок 1.8 – Головна сторінка вебсайту «АВТОНОМЕР»

Сайт дозволяє отримувати інформацію з різних баз даних, що дозволяє отримати повну інформацію про транспортний засіб (рис. 1.9). Але є платним сервісом, та запитує про реєстрацію. Без оплати ми не зможемо побачити інформацію про VIN-номер.

До переваг можна віднести:

- робота з багатьма базами даних;
- можливість реєстрації;
- є додатки для мобільних пристроїв;
- є інформація про недавно вкрадені транспортні засоби;
- є інформація про транспортні засоби які часто шукають;
- є інформація про транспортні засоби які ми шукали раніше.

Недоліки:

- сервіс є частково платним;
- повільно працює;
- має застарілий дизайн.

1.4 Визначення вимог до інформаційної системи

Для здійснення подальшого проектування вебдодатку та програмної реалізації необхідно визначити функціональні вимоги:

- зручний та привабливий інтерфейс вебзасобу для користувачів;
- пошук повинен працювати швидше 1 секунди;
- час, який займе у користувача на перший запуск та пошук, не повинен перевищувати 20 секунд;
- вебзастосунок повинен працювати без збоїв.

Метою даної дипломної роботи є створення вебзастосунку, яку може використовувати будь-який користувач з телефоном або комп'ютером з встановленим Інтернет-браузером без необхідності встановлення додаткового програмного забезпечення.

Рисунок 1.9 – Повна інформація про транспортний засіб з вебсайту «АВТОНОМЕР»

Вебзастосунок повинен надати всю необхідну інформацію про транспортний засіб щоб зрозуміти чи вкрадений автомобіль:

- марка;
- модель;
- рік;
- колір;
- VIN-номер.

Вимоги до надійності:

- програмний продукт повинен видавати повідомлення при некоректному заповненні числових та текстових полів;
- програмний продукт повинен видавати повідомлення при некоректній авторизації;
- при неправильному пошуку видавати повідомлення.

Вебзастосунок повинен отримувати ці данні від відкритих даних та постійно оновлюватись.

1.5 Висновок

В даному розділі було досліджено предметну область, а саме визначені основні поняття і терміни, пов'язані з предметною областю і об'єктом розробки, та характеристики комплексу задач по створенню вебзастосунку для швидкого пошуку транспортних засобів у розшуку.

Було сформульовано і визначено завдання до роботи.

2 ОПИС ПРОГРАМИ

Розвиток мережі Інтернет призвело до появи нової категорії програм – Web-додатків. До Web-додатків відносять набір Web-сторінок, сценаріїв і інших програмних засобів, розташованих на одному або декількох комп'ютерах (клієнтських і серверних) і об'єднаних для виконання прикладної задачі. При цьому Web-додатки, що публікують бази даних в Інтернет, являють собою окремий клас Web-додатків. Сучасні інформаційні системи, побудовані на основі Web-додатків, що використовують бази даних, базуються на багаторівневої клієнт-серверній архітектурі і принципах функціонування Інтернету [8].

Web-додатки – це особливий тип програм, які побудовані за архітектурою «клієнт-сервер». Особливість їх полягає в тому, що саме Web-додаток знаходиться і виконується на сервері, а клієнт при цьому отримує тільки результати роботи. Робота вебдодатку ґрунтується на отриманні запитів від користувача (клієнта), їх обробці та видачі результату. Передача запитів від клієнта до сервера та отримання результатів їх обробки відбувається через мережу Інтернет.

2.1 Технології створення web-додатків

Сучасні Web-додатки стають все більш складними. Сучасний вебдодаток – це унікальна розробка, орієнтована на вирішення певних завдань. Воно стає все більш затребуваним, оскільки дозволяє ефективніше вести бізнес-процеси.

Будь-який подібний додаток відображається на екрані монітора або смартфона, як вебсторінка. На ній міститься не тільки певна інформація, але й передбачені функціональні кнопки, вікна. Всі вони дозволяють виконувати конкретні дії для отримання необхідного результату.

Позитивна особливість розробки – клієнтові немає необхідності встановлювати на свій пристрій програмне забезпечення. Платформа працює безпосередньо у браузері [9].

Для розробки серверної частини використовуються різні мови програмування:

- Ruby;
- Java;
- Php;
- Python;
- JavaScript.

Розробка клієнтської частини передбачає використання:

- JavaScript;
- HTML;
- CSS.

Для розробки вебзастосунку для пошуку транспортного засобу у розшуку була вибрана мова програмування JavaScript, бібліотека Next.js, а також середовище Visual Studio Code.

JavaScript – динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв вебсторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд вебсторінки [10].

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу [10].

Мова JavaScript використовується для:

- написання сценаріїв вебсторінок для надання їм інтерактивності;
- створення односторінкових вебзастосунків (ReactJS, AngularJS, Vue.js);
- програмування на стороні сервера (Node.js);
- стаціонарних застосунків (Electron, NW.js);
- мобільних застосунків (React Native, Cordova);
- сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite чи Apache JMeter);
- всередині PDF-документів.

2.2 Вибір мови програмування для серверної частини

Перед розробкою будь-якого програмного забезпечення необхідно провести огляд існуючих технологій та мов програмування, проаналізувати доречність їх використання у розробці. Серверна частина отримує запити від клієнтської частини, опрацьовує їх, виконує запити до бази даних, та реалізує логіку програми.

Найпоширенішими мовами програмування для розробки серверної частини на початок 2022 року є:

- Java;
- PHP;
- C#;
- Python;
- Go;
- Ruby;
- TypeScript;
- JavaScript;
- C++ [11].

Відповідно до вимог, поставлених до програмного забезпечення проаналізуємо мови JavaScript та Java (табл. 2.1). На початок 2022 року найпопулярнішими фреймворками для розробки серверної частини вебзастосунків мовою Java є Spring та Hibernate [12], а для розробки мовою JavaScript – Node.js.

Таблиця 2.1 – Порівняння мов програмування

Параметр	PHP	Node.js	Java
Сфера застосування	Веброзробка (сервер)	Веброзробка (сервер)	Веброзробка (сервер), мобільні та desktop додатки
Наявність CMS	Багато	Небагато	Небагато
Швидкодія	Висока	Середня	Висока
Вартість	Безкоштовний	Безкоштовний	Безкоштовний
Паралельна робота	Багатопоточність	Асинхронність одного потоку	Багатопоточність
SSR	+	+	+

Головними пунктами для порівняння цих фреймворків, як серверних, є швидкодія та підтримка високої завантаженості: як швидко застосунок відповідає на дію користувача. Порівнюючи швидкість Java із Node.js, стає зрозуміло, що Node.js є більш оптимальним та швидким рішенням лише при досить значній кількості запитів. Це пояснюється тим, що Node.js працює асинхронно не блокуючи потік вводу даних, а також використовує потужний двигун chrome V8. Саме тому для написання чатів, ботів, вебдодатків, що працюють у реальному часі краще підходить Node.js. Але для створення повноцінного серверу необхідно встановлювати додаткові пакети із npm, у той же час у Spring ми маємо всі необхідні інструменти.

Наступним аспектом для порівняння мов Node.js та Java є масштабованість та паралельна робота вебдодатків. Зазвичай під масштабованістю розуміють стабільну роботу вебдодатків зі збільшенням кількості одночасних запитів. Node.js є асинхронним та однопоточним, завдяки чому операції вводу та виводу не блокують потік виконання,

завершуючись за його межами. Вебфреймворки написані мовою Java, у свою чергу, працюють у режимі багатопоточності, що дозволяє використовувати повністю ресурси сервера, а також можливо додати певні пакети, що дадуть змогу обробляти асинхронні дані. Таким чином, мова програмування Java має повний інструментарій для досягнення необхідної масштабованості розроблюваного програмного забезпечення.

Для програмного продукту було обрано Node.js, виходячи з вимог та умов програмного продукту.

2.3 Обґрунтування вибору клієнтської частини

Для написання клієнтською частини використовують мову програмування JavaScript. Для того щоб писати вебсайти легко та швидко використовують фреймворки та бібліотеки. Найпопулярніші фреймворки:

- React.js;
- Vue.js;
- Angular.js;
- Svelte.

Таблиця 2.2 – Порівняння фреймворків

Параметр	React.js	Vue.js	Angular.js	Svelte
Фреймворк / бібліотека / компілятор	бібліотека	фреймворк	фреймворк	компілятор
Віртуальний DOM	+	+	Incremental DOM	-
Шаблон проектування	V	MVVM	MVC	-
Поріг входження	низький	низький	великий	низький
Розмір готового застосунку	середній	маленький	великий	маленький

React – відкрита JavaScript бібліотека для створення інтерфейсів користувача, яка покликана вирішувати проблеми часткового оновлення

вмісту вебсторінки, з якими стикаються в розробці односторінкових застосунків. Розробляється Meta (раніше Facebook) і спільнотою індивідуальних розробників [13].

React дозволяє розробникам створювати великі вебзастосунки, які використовують дані, котрі змінюються з часом, без перезавантаження сторінки. Його мета полягає в тому, щоб бути швидким, простим, масштабованим. React обробляє тільки користувацький інтерфейс у застосунках. Це відповідає видові у шаблоні модель-вид-контролер (MVC), і може бути використане у поєднанні з іншими JavaScript бібліотеками або в великих фреймворках MVC, таких як AngularJS. Він також може бути використаний з React на основі надбудов, щоб піклуватися про частини без користувацького інтерфейсу побудови вебзастосунків. Як бібліотеку інтерфейсу користувача React найчастіше використовують разом з іншими бібліотеками, такими як Redux [13].

Плюси використання React:

- легкий перехід між версіями бібліотеки;
- легко масштабується;
- швидкий час роботи;
- має низький поріг входження;
- має велику кількість готових компонентів;
- використовує Virtual DOM, для оптимізації оновлення DOM дерева, але витрачає для цього велику кількість ресурсів;
- велика гнучкість та свобода роботи з веб компонентами.

Недоліки:

- не має повного MVC, тільки View, але використовуючи бібліотеки можливо легко використовувати як MVC;
- для початку роботи потрібно збирати фреймворк, з великої кількості бібліотек.

Vue.js – JavaScript-фреймворк що використовує шаблон MVVM для створення інтерфейсів користувача на основі моделей даних, через реактивне зв'язування даних [14].

Має такі плюси:

- має легку підтримку, та легкий перехід між версіями бібліотеки;
- легко масштабується;
- має маленький розмір готового вебзастосунку;
- використовує Virtual DOM;
- швидкий час роботи;
- низький поріг входу;
- має різні переходи та роутинг за замовчуванням.

Також такі недоліки:

- менш гнучкий компонентний підхід;
- важкість керування життєвим циклом;
- перехід з однієї версії до іншої іноді потребує повного переписування компонентів;
- важкість відлагодження та пошуку проблем.

Можна зробити висновок, що Vue.js більше підходить для розробки більшості застосунків, але якщо треба робити складне відображення, то React підійде більше.

Angular – написаний на TypeScript фреймворк з відкритим кодом, який розробляється під керівництвом Angular Team у компанії Google, а також спільнотою приватних розробників та корпорацій. Angular – це AngularJS, який був переосмислений та перероблений тією ж командою розробників [15].

Переваги:

- повноцінний MVC;
- двостороннє зв'язування даних;
- легкість тестування;

- чітка та жорстка структура, завдяки чому є стабільним та надійним, тому рідко використовується для маленьких проектів.

Недоліки:

- великий розмір самого фреймворку, який у 7-8 разів більший за Vue.js чи React;

- великий поріг входження;
- іноді важка міграція між версіями;
- уповільнення роботи під час використання;
- потребує великих ресурсів.

Svelte – це безкоштовний компілятор інтерфейсу з відкритим вихідним кодом, створений Річем Харрісом і підтримується членами основної команди Svelte. Svelte не є монолітною бібліотекою JavaScript, імпортованою програмами: натомість Svelte компілює шаблони HTML у спеціалізований код, який безпосередньо маніпулює DOM, що може зменшити розмір переданих файлів і покращити продуктивність клієнта; код програми також обробляється компілятором, вставляючи виклики для автоматичного переобчислення даних і повторного візуалізації елементів UI, коли дані, від яких вони залежать, змінюються. Це також дозволяє уникнути накладних витрат, пов'язаних з часом виконання, такі як віртуальний DOM, на відміну від традиційних фреймворків (таких як React і Vue), які виконують основну частину своєї роботи під час виконання, тобто в браузері. Сам компілятор написаний на TypeScript. Його вихідний код ліцензований за ліцензією MIT і розміщений на GitHub [16].

До переваг можна віднести:

- має маленький розмір готового вебзастосунку, бо компілює вихідні файли;
- менше навантаження на систему;
- простота використання для новачків;
- велика продуктивність;
- точково оновлює DOM.

До недоліків слід віднести:

- великий поріг входження;
- складно повторно використовувати компоненти;
- небагато готових компонентів.

З цього можна зробити висновок, що Svelte більше підходить для маленьких компонентів, чи вебзастосунків.

Виходячи з даних вимог та умов, найбільш оптимальним інструментом для розробки обрано бібліотека React.

2.4 Аналіз вибору СКБД

В даний час поширені два типи СКБД: реляційні та нереляційні, звані SQL та NoSQL відповідно. Розглянемо, чим відрізняються реляційні та нереляційні СКБД з урахуванням структури, продуктивності, масштабованості та безпеки [17].

Реляційна база даних – це тип сховища даних, що впорядковує дані таблиці, пов'язані між собою. Структурована мова запитів є основою цих систем, оскільки використовується для зв'язку та управління. СКБД мають зумовлену схему, тобто дані зберігаються у рядках (записах) та стовпцях (атрибутах) із суворою структурою. Кожен запис зазвичай містить значення для кожного атрибута, що призводить до чітких залежностей між різними точками даних [17].

Реляційні бази зазвичай масштабуються по вертикалі, що означає, що дані зберігаються одному сервері, а масштабування виконується шляхом додавання більшої кількості ресурсів [17].

Реляційні бази даних демонструють високу продуктивність при інтенсивних операціях читання/запису з невеликими та середніми наборами даних. Вони також забезпечують підвищену швидкість пошуку даних за рахунок додавання індексів поля даних для запитів і об'єднання таблиць.

Однак у разі збільшення обсягу даних та запитів користувачів продуктивність може знизитися.

Нереляційна база даних – це таблична база даних, яка використовує різні моделі даних для зберігання та управління. Тут, на відміну більшості традиційних СКБД, не використовується таблична схема рядків і стовпців, а застосовується модель, оптимізована під конкретні вимоги типу даних, що зберігаються. Наприклад, дані можуть зберігатися як прості пари "ключ - значення", документи JSON або граф, що складається з ребер та вершин [17].

Коли кількість даних та запитів збільшується, нереляційні бази даних зазвичай масштабуються горизонтально шляхом додавання до пулу додаткових серверів. Вони обмінюються даними між різними серверами, кожен із яких містить лише частина даних, зменшуючи частоту запитів на секунду кожному сервері [17].

Нереляційні БД відомі своєю високою продуктивністю, забезпечуючи одночасний доступ до великої кількості користувачів. Можуть зберігати необмежену кількість наборів всіх типів і форм. Вони також досить гнучкі, коли справа доходить до зміни типів даних. СКБД NoSQL мають відносно низьку безпеку, що є негативним фактором для багатьох інфраструктур.

Найпопулярніші системи баз даних:

- MySQL;
- PostgreSQL;
- SQLite;
- MongoDB;
- Microsoft SQL Server;
- Redis;
- MariaDB;
- Firebase;
- Elasticsearch;
- Oracle.

Таблиця 2.3 – Порівняння характеристик СКБД

Параметр	Тип бази даних	Легкість документації	Масштабованість	Легкість навчання
MySQL	SQL	середня	вертикальна, складна	середня
Maria DB	SQL	легка	вертикальна	середня
Oracle	Multi-model, SQL	легка	вертикальна	складна
PostgreSQL	Object-relational, SQL	середня	вертикальна	складна
MSSQL Server	T-SQL	легка	вертикальна, складна	складна
MongoDB	NoSQL, document-oriented	легка	горизонтальна	середня
Redis	NoSQL, key-value	легка	горизонтальна	середня
Elasticsearch	NoSQL, document-oriented	середня	горизонтальна	складна
Firebase	NoSQL, real-time database	легка	горизонтальна	середня

Розглянемо деякі з них. MySQL – вільна система керування реляційними базами даних, яка була розроблена компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL – одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних вебсторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

Має такі переваги:

- безкоштовний;
- простий синтаксис та помірна складність;
- сумісність із хмарними рішеннями.

Недоліки:

- проблеми масштабованості;
- не повністю відкритий вихідний код;
- обмежена відповідність стандартам SQL.

MariaDB, безкоштовне відгалуження від MySQL з відкритим вихідним кодом має комерційну підтримку. Спочатку розроблялося як СКБД, сумісна на 100% з MySQL. Поширюється під загальною публічною ліцензією GNU та має аналогічні команди, API та бібліотеки, як і MySQL [17].

Переваги:

- шифрування;
- широка функціональність;
- висока продуктивність.

До недолік можна віднести:

- на реальних проектах показала себе недостатньо стабільною та надійною СКБД, явно поступаючись MySQL;
- молода спільнота;
- різниця між версіями оновлень MySQL та MariaDB.

MongoDB – документо-орієнтована система керування базами даних з відкритим вихідним кодом, яка не потребує опису схеми таблиць. MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів [18].

MongoDB підтримує зберігання документів в JSON-подібному форматі, має досить гнучку мову для формування запитів, може створювати індекси для різних збережених атрибутів, ефективно забезпечує зберігання великих бінарних об'єктів, підтримує журналювання операцій зі зміни і додавання даних в БД, може працювати відповідно до парадигми Map/Reduce, підтримує реплікацію і побудову відмовостійких конфігурацій. У MongoDB є вбудовані засоби із забезпечення шардінгу (розподіл набору даних по серверах на основі певного ключа), комбінуючи який з реплікацією

даних можна побудувати горизонтально масштабований кластер зберігання, в якому відсутня єдина точка відмови (збій будь-якого вузла не позначається на роботі БД), підтримується автоматичне відновлення після збою і перенесення навантаження з вузла, який вийшов з ладу. Розширення кластера або перетворення одного сервера на кластер проводиться без зупинки роботи БД простим додаванням нових машин [18].

Переваги:

- документо-орієнтоване сховище (проста та потужна JSON-подібна схема даних);
- досить гнучка мова для формування запитів;
- динамічні запити;
- повна підтримка індексів;
- профілювання запитів;
- швидкі оновлення «на місці»;
- ефективне зберігання бінарних даних великих обсягів, наприклад, фото та відео;
- журналювання операцій, що модифікують дані в БД;
- підтримка відмовостійкості і масштабованості: асинхронна реплікація, набір реплік і шардінг;
- має зрозумілу документацію;
- може працювати відповідно до парадигми MapReduce.

Недоліки:

- транзакції з використанням MongoDB є складними;
- MongoDB не настільки відповідає вимогам ACID (атомарність, узгодженість, ізолюваність та стійкість), як реляційні бази даних;
- у MongoDB немає положень про процедури або функції, що зберігаються, тому не вдається реалізувати будь-яку бізнес-логіку на рівні бази даних, що можна зробити в реляційних БД.

Виходячи з даних вимог та умов, найбільш оптимальним системою керування базою даних для даного програмного продукту обрано MongoDB.

2.5 Висновок

В даному розділі було проведено аналіз та вибір інструментарію розробки програмного продукту. Проведено порівняльну характеристику мов програмування та встановлено, що найбільші переваги для розробки серверної та клієнтської частин має мова програмування JavaScript. В якості фреймворків для серверної частини Node.js та для клієнтської бібліотека React.js. Системи зберігання даних обрано СКБД MongoDB.

3 РОЗРОБКА ПРОГРАМИ

3.1 Загальний опис системи

Розроблюване програмне забезпечення у вигляді вебзастосунку призначене для пошуку автомобілів у розшуку. Воно має забезпечити отримання достовірної інформації про транспортний засіб без затримки. Вебзастосунок завантажує данні з сайту відкритих даних та оновлює їх у своїй базі даних, для подальшого пошуку.

Для більш швидкого завантаження сайту використано SSR. Server-Side Rendering (SSR) – це рендеринг клієнтського або універсального додатка в HTML на сервері. Він генерує повний HTML для сторінки на сервері у відповідь на запит або перехід по посиланню. Це дозволяє уникнути додаткових запитів даних, наповнення шаблонів на сторінці клієнта, оскільки вони обробляються до того, як браузер отримує відповідь [19].

3.2. Архітектура системи

При розробці вебзастосунку було використано монолітну архітектуру та фреймворк Next.js. Next.js – це відкритий JavaScript фреймворк, зроблений на основі React.js, для створення вебзастосунків. Фреймворк було призначено для рішення проблеми React.js, зв'язану із відображенням застосунку на стороні сервера – SSR [20].

При використанні Next.js є можливість створювати серверну частину за допомогою відкритого JavaScript оточення – Node.js. Фреймворк реалізований як і відкрите програмне забезпечення під ліцензією MIT та простою для розуміння документацією [20].

По своїй суті архітектура моноліт являє собою декілька технологій в одному місці. Вони надають REST API для використання головним сервером. На рис. 3.1. наведене схематичне зображення архітектури застосунку. Даний вебзастосунок загалом застосовує клієнт-серверну архітектуру. Клієнт-

серверна архітектура – це розподілена структура додатків, яка розбивається на:

- серверну частину, яка надає інформацію;
- клієнтську частину, яка використовує сервіси, що надаються серверами;
- мережна частина, яка забезпечує взаємодію між клієнтами та серверами.

В дипломному проєкті клієнт та сервер пов'язані між собою за допомогою REST API.

Клієнт використовуючи REST API робить запити на сервер, який в свою чергу виконує певні дії, що диктує запит, на надсилає відповідь клієнту.

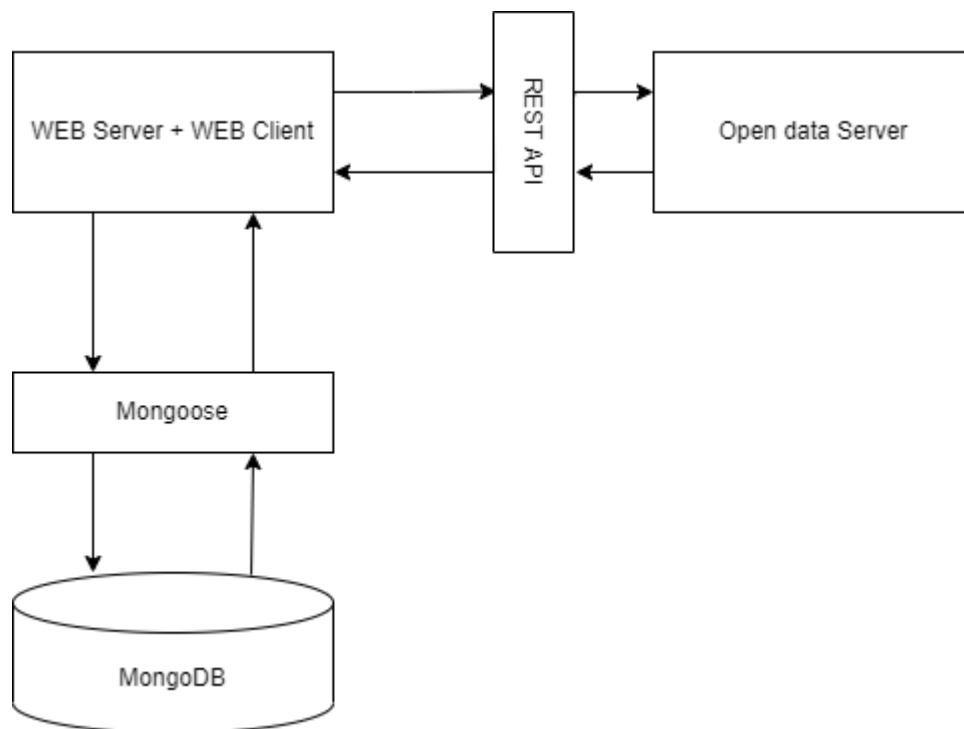


Рисунок 3.1 – Схематичне зображення архітектури системи

Для розробки серверної частини застосунку було обрано СКБД MongoDB та середовище Node.js. Для їх взаємодії було використано бібліотеку mongoose. Це офіційний драйвер MongoDB для Node.js. Надає API високого рівня поверх ядра mongodb, призначеного для кінцевих

користувачів. Схема бази даних Для реалізації розроблюваного застосунку було створено таблицю Cars.

Таблиця Cars містить у собі список інформації про вкрадені автомобілі. Ця інформація буде синхронізуватись з сайтом відкритих даних для подальшого використання. Більш детальна інформація про назви полів та їх типи наведена у табл. 3.1.

Таблиця 3.1 – Таблиця Cars

Назва поля	Тип даних	Додаткова інформація
ID	String	Ідентифікаційний номер
ODV	String	Відділення поліції
BRAND	String	Бренд
COLOR	String	Колір
VEHICLENUMBER	String	Номер реєстрації транспортного засобу
CLASSISNUMBER	String	Номер шасі
BODYNUMBER	String	VIN-номер транспортного засобу
ENGINENUMBER	String	Номер двигуна
THEFT_DATA	String	Дата крадіжки
INSERT_DATE	Date	Дата добавляння в базу даних

3.3 Висновок

В даному розділі було проведено аналіз та вибір інструментарію розробки ППІ та було спроектовано архітектуру майбутнього ППІ.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Опис застосування програми

Вебзастосунок призначен для отримання інформації про транспортні засоби у розшуку та має наступні функції:

- пошук транспортного засобу у розшуку по державному номеру;
- пошук транспортного засобу у розшуку по VIN-номеру;
- відображення даних про транспортні засоби у розшуку;
- автоматичне відновлення даних.

4.2 Керівництво програміста

Мінімальні та необхідні системні вимоги до апаратної платформи ПП:

- мінімум 2 ГБ оперативної пам'яті, рекомендується 4 ГБ оперативної пам'яті;
- вільне місце на диску 5 ГБ мінімум;
- база даних: MongoDB;
- Windows, Linux, macOS.

Також необхідно встановити такі програми:

- Node.js;
- середовище розробки (Visual Studio Code, JetBrains WebStorm, або інше але з підтримкою веброзробки);
- MongoDB;
- Postman;
- веббраузер (Google Chrome, Mozilla Firefox, Safari, Opera, Chromium, або інший).

Для старту проекту необхідно встановити пакети, для цього виконати команду `npm i`. Потім виконати команду `npm run dev`. А потім перейти на посилання.

З сайту відкритих даних данні оновлюються автоматично після запуску, а також кожні 3 години.

4.2.1 Звернення до програмного продукту (файли проекту)

У корені знаходяться наступні папки та файли:

- components – містить файли React компонентів;
- pages – містить файли сторінок;
- public – містить файли для публікації;
- server – містить файли серверної частини;
- services – містить файли сервісів, допоміжних функцій;
- styles – містить файли стилів;
- .eslintrc.json – файл конфігурації eslint;
- .gitignore – файл конфігурації файлів та папок які не будуть зафіксовані у комміті;
- next.config.js – файл конфігурації Next.js;
- package.json – файл конфігурації залежності пакетів;
- package-lock.json – файл зафіксованих залежності пакетів;
- README.md – файл рекомендацій по встановленню та роботи з ПП.

4.3 Керівництво користувача

Для функціонування розробленого програмного продукту необхідно мати пристрій з встановленим браузером та Інтернет зв'язок.

Для роботи з сайтом необхідно перейти за посиланням. Після переходу за посиланням одразу буде відкрита сторінка для пошуку за реєстраційним або VIN-номером транспортного засобу (рис 4.1).

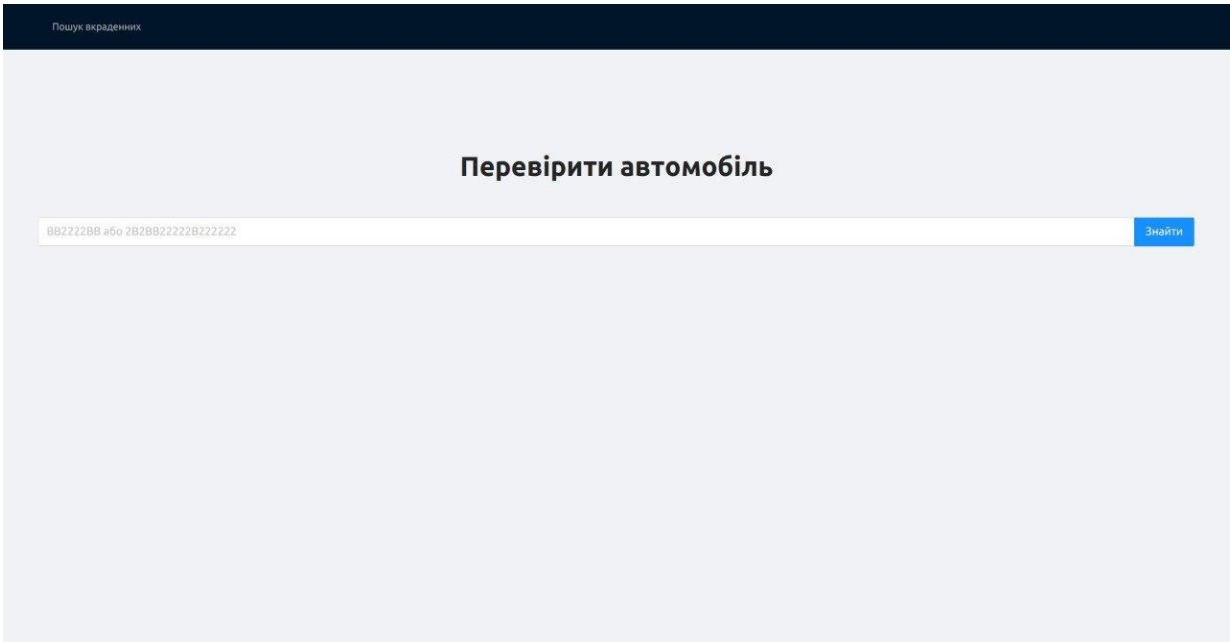


Рисунок 4.1 – Головна сторінка сайту, та сторінка пошуку за реєстраційним або VIN-номером транспортного засобу

Після вводу реєстраційного або VIN-номеру та натискання на кнопку знайти, якщо транспортний засіб буде у базі вкрадених автомобілів, то нас перекине на сторінку інформації про транспортний засіб у розшуку (рис 4.2).

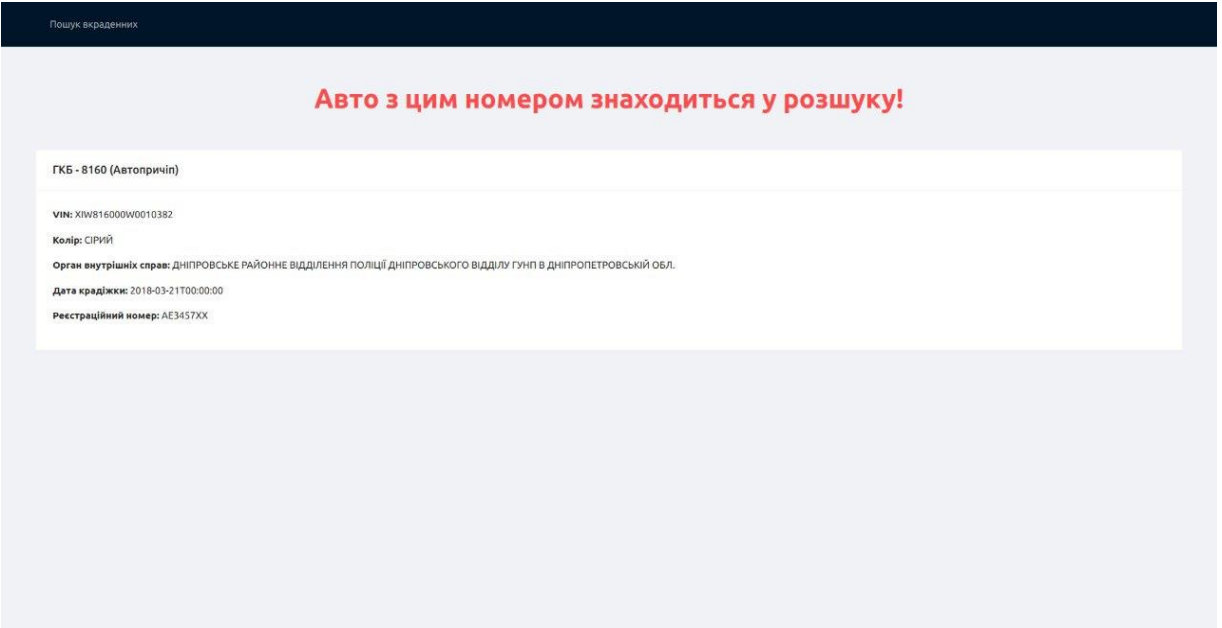


Рисунок 4.2 – Сторінка інформації про транспортний засіб у розшуку

Якщо за введеним номером не буде інформації у базі вкрадених автомобілів, то буде відображена сторінка з інформацією, що інформацію про транспортний засіб не знайдено (рис 4.3).

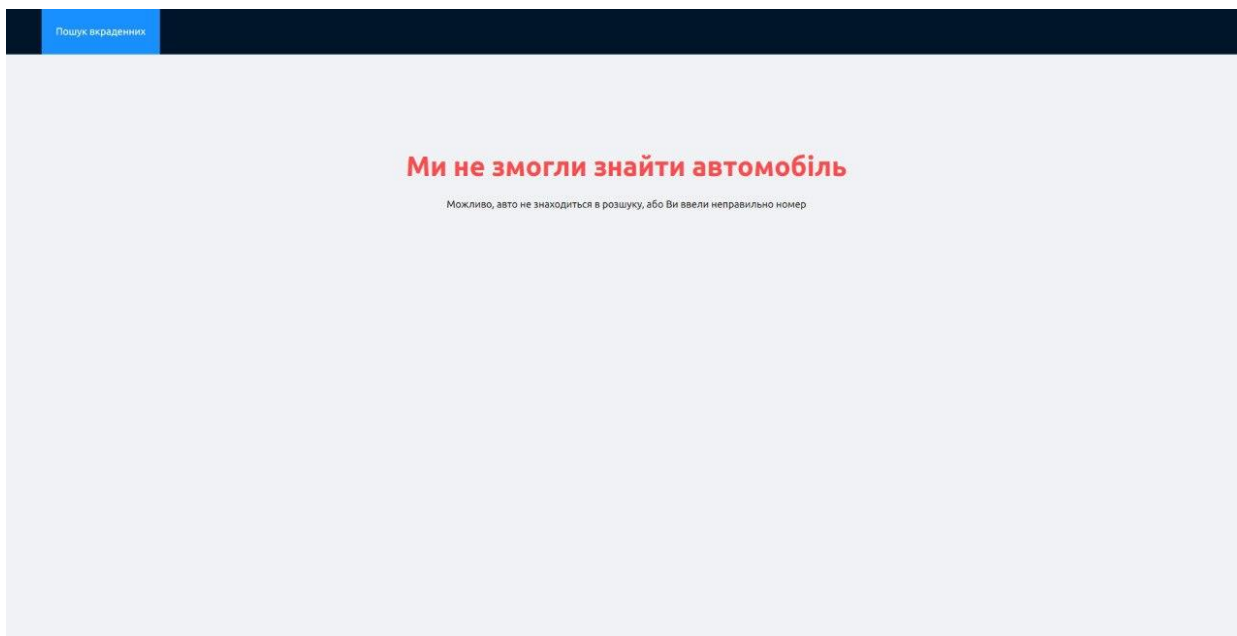


Рисунок 4.3 – Сторінка інформацію про не знайдений транспортний засіб у базі вкрадених транспортних засобів

4.4 Висновок

В даному розділі було описано призначення розробленого ПП та мінімально необхідні апаратна платформа для його встановлення та використання. Була наведена інструкція по роботі оператора з ПП. Інтерфейс користувача розробленого ПП є простим та інтуїтивно зрозумілим і не потребує додаткового навчання оператора.

ВИСНОВКИ

В результаті виконання дипломної роботи був розроблений додаток, призначений для пошуку транспортних засобів у розшуку. Додаток має наступні функції:

- відображення даних про транспортні засоби у розшуку;
- автоматичне відновлення даних;
- пошук по державному номеру транспортного засобу;
- не обов'язкова реєстрація користувачів, для зберігання пошуку.

Програмний продукт було розділено на клієнт та сервер за допомогою клієнт-серверної архітектури Rest API. Клієнт був розроблений за допомогою HTML, CSS та JavaScript-фреймворку React.js. Серверна частина була розроблена за допомогою серверу Node.js та пакету Webpack-Dev-Server, базою даних MongoDB та бібліотекою для роботи з базою даних Mongoose.

Інформація для бази даних було взято з сайту відкритих даних data.gov за допомогою API запитів.

Програмне забезпечення розроблене за допомогою візуального середовища розробки Visual Studio Code.

Програмний продукт працює на персональному комп'ютері та на смартфонах. Додаток має зручний, максимально орієнтований на некваліфікованого кінцевого користувача інтерфейс.

Програмний засіб був протестований та виявився працездатним, має достатньо велику швидкість роботи.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Сайт [Електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82>
2. Транспортний засіб [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BF%D0%BE%D1%80%D1%82%D0%BD%D0%B8%D0%B9_%D0%B7%D0%B0%D1%81%D1%96%D0%B1
3. Статистика викрадення авто 2021 в Україні, або Підсумки року від StarLine [Електронний ресурс]. – Режим доступу: <https://starline.in.ua/statistika-ugona-avto-2021-v-ukraine-ili-itogi-goda-ot-starline.html>
4. Викрадення авто під час карантину в Україні. Як змінилася статистика угонів у 2021 році? [Електронний ресурс]. – Режим доступу: <https://benishgps.com/uk-ua/blog/avtougony-pid-chas-karantynu-yak-zminylasya-statystyka-avtougony-v-ukrayini/>
5. Викрадення. Особливості відстійників у різних містах [Електронний ресурс]. – Режим доступу: <https://www.ugona.net/news/ugony-osobennosti-otstoinikov-v-raznykh-gorodakh-1364.html>
6. Відкриті дані [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%B4%D0%BA%D1%80%D0%B8%D1%82%D1%96_%D0%B4%D0%B0%D0%BD%D1%96
7. Про внесення змін до деяких законів України щодо доступу до публічної інформації у формі відкритих даних [Електронний ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/319-19#Text>
8. Web-додатки для публікації баз даних [Електронний ресурс]. – Режим доступу: <https://uadoc.zavantag.com/text/26921/index-5.html>
9. Вебдодаток і його характеристики [Електронний ресурс]. – Режим доступу: <https://www.centum-d.com/veb-dodatok-yogo-harakteristiki>
10. JavaScript [Електронний ресурс]. – Режим доступу:

<https://uk.wikipedia.org/wiki/JavaScript>

11. Рейтинг мов програмування 2022 [Електронний ресурс]. – Режим доступу: <https://dou.ua/lenta/articles/language-rating-2022/>

12. Best Java Frameworks to Use in 2022 [Electronic resource]. - Access mode: <https://hackr.io/blog/java-frameworks>

13. React [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/React>

14. Vue.js [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Vue.js>

15. Angular (фреймворк) [Електронний ресурс]. – Режим доступу: [https://uk.wikipedia.org/wiki/Angular_\(%D1%84%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA](https://uk.wikipedia.org/wiki/Angular_(%D1%84%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA)

16. Svelte [Electronic resource]. - Access mode: <https://en.wikipedia.org/wiki/Svelte>

17. Аналіз популярних реляційних систем керування базами даних (2022) [Електронний ресурс]. – Режим доступу: <https://drach.pro/blog/hi-tech/item/196-popular-relational-dbms-2022>

18. MongoDB [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/MongoDB>

19. Рендеринг: різновиди, застосування, порівняльна характеристика [Електронний ресурс]. – Режим доступу: <https://medium.com/@ralabs.ua/%D1%80%D0%B5%D0%BD%D0%B4%D0%B5%D1%80%D0%B8%D0%BD%D0%B3-%D1%80%D1%96%D0%B7%D0%BD%D0%BE%D0%B2%D0%B8%D0%B4%D0%B8-%D0%B7%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F-%D0%BF%D0%BE%D1%80%D1%96%D0%B2%D0%BD%D1%8F%D0%BB%D1%8C%D0%BD%D0%B0-%D1%85%D0%B0%D1%80%D0%B0%D0%BA%D1%82%D0%B5%D1%80%>

D0%B8%D1%81%D1%82%D0%B8%D0%BA%D0%B0-cc12e3a6707d

20. Next.js [Электронный ресурс]. – Режим доступа:
<https://ru.wikipedia.org/wiki/Next.js>

ДОДАТОК А
Технічне завдання

Вступ

Розроблюваний програмний продукт використовується для вирішення проблеми повільного пошуку транспортних засобів у розшуку.

ПП забезпечує процес пошуку транспортних засобів у розшуку.

A.1 Мета розробки і її призначення

Об'єкт дослідження – програмні засоби для пошуку транспортних засобів у розшуку.

Предмет дослідження – вебсайт для пошуку транспортних засобів у розшуку.

Мета роботи – створення вебзастосунку для швидкого пошуку транспортних засобів у розшуку.

Актуальність роботи полягає в тому, що останнім часом кількість вкрадених автомобілів збільшується.

A.2 Підстави для розробки

Підставою для виконання розробки є завдання на дипломний проєкт на тему “Розробка вебсайту крамниці канцелярських товарів”, затверджене наказом по НУ «Запорізька політехніка» № 102 від 27 квітня 2022 року.

A.3 Основні вимоги до програми, що розробляється, початковим даним і результату

A.3.1 Вимоги до функціональних характеристик

Вимоги до функціональних характеристик:

- зручний та привабливий інтерфейс веб-засобу для користувачів;
- пошук повинен працювати швидше 1 секунди;

- час, який займе у користувача на перший запуск та пошук, не повинен перевищувати 20 секунд;
- веб-застосунок повинен працювати без збоїв.

A.3.2 Вимоги до надійності

ПП повинна виконувати наступні вимоги до надійності:

- захищати ПП від помилкових дій користувача (наприклад користувач ввів невалідні дані);
- обробляти виняткові ситуації (наприклад, відсутність підключення до бази даних).

ПП повинен нормально функціонувати при безперервній роботі комп'ютера. У разі виникнення збоїв в роботі апаратури, відновлення нормальної роботи ПП повинно проводитися після перезавантаження системи.

A.3.3 Умови експлуатації

Умови експлуатації:

- програмне забезпечення не потребує обслуговуючого персоналу;
- для користування програмним забезпеченням не потрібна додаткова кваліфікація.

A.3.4 Вимоги до складу та параметрів технічних засобів

Вимоги до складу та параметрів технічних засобів:

- мінімум 2 ГБ оперативної пам'яті, рекомендується 4 ГБ оперативної пам'яті;
- вільне місце на диску 5 ГБ мінімум;
- база даних: MongoDB;

- Windows, Linux, macOS.

A.3.5 Вимоги до маркування та упакування

Програмне забезпечення не потребує додаткових дій при маркуванні та упакуванні.

A.3.6 Вимоги до транспортування та збереження

Вимоги до транспортування й зберігання аналогічні вимогам, які пропонуються до носія, на якому записана програма.

A.4 Вимоги до програмної документації

Склад програмної документації повинен включати в себе:

- технічне завдання;
- керівництво програміста;
- керівництво оператора;
- опис програми.

A.5 Основні обмеження на установку і використання програмного продукту

Розроблюваний програмний продукт потребує інсталяції. Програмний продукт повинен використовуватись на комп'ютерах, які задовільняють вимогам та параметрам технічних засобів. Програмний продукт може використовуватись користувачем у будь-який момент часу, не конфліктуючи з іншими програмами.

ДОДАТОК Б
Текст файлів програми

Б.1 Файл FindCarForm.jsx

```
const FindCarForm = ({ onSubmit }) => {
  return (
    <div style={{ padding: '25px 0'}}>
      <Search
        placeholder="BB2222BB або 2B2BB22222B222222"
        enterButton="Знайти"
        size="large"
        onSearch={onSubmit}
      />
    </div>
  );
}
export default memo(FindCarForm)
```

Б.2 Файл index.jsx

```
const { Title } = Typography;
const FindCard = () => {
  const router = useRouter();
  const handleSubmit = carNumber => {
    router.push(`wanted-car/${carNumber}`);
  }
  return (
    <div
      style={{
        paddingTop: '15vh',
      }}
    >
      <Typography>
        <Title
          style={{ padding: '5px 0', textAlign: 'center' }}
        >
          Перевірити автомобіль
        </Title>
      </Typography>
      <FindCarForm
        onSubmit={handleSubmit}
      />
    </div>
  );
}
```

```

    </div>
  )
}
export default memo(FindCard)

```

Б.3 Файл CarTextItem.jsx

```

const { Paragraph, Text } = Typography;
const CarTextItem = ({ label, text }) => {
  if (!text) {
    return null
  }
  const labelText = label ? `${label}: ` : "";
  return (
    <Typography>
      <Paragraph>
        <Text strong style={{ minWidth: '250px' }}>
          {labelText}
        </Text>
        {text}
      </Paragraph>
    </Typography>
  )
}
export default memo(CarTextItem);

```

Б.4 Файл index.jsx

```

const { Title, Text } = Typography;
const FoundedCars = ({
  BODYNUMBER,
  BRAND,
  CHASSISNUMBER,
  COLOR,
  ENGINENUMBER,
  OVD,
  THEFT_DATA,
  VEHICLENUMBER,
}) => (
  <div>

```

```

<Typography
  style={{
    paddingTop: '50px',
  }}
>
  <Title
    style={{ textAlign: 'center' }}
    type="danger"
  >
    Авто з цим номером знаходиться у розшуку!
  </Title>
</Typography>
<Card
  title={BRAND}
  bordered={false}
  style={{ marginTop: '50px' }}
>
  <CarTextItem
    label="VIN"
    text={BODYNUMBER}
  />
  <CarTextItem
    label="Номер шасі"
    text={CHASSISNUMBER}
  />
  <CarTextItem
    label="Колір"
    text={COLOR}
  />
  <CarTextItem
    label="Номер двигуна"
    text={ENGINENUMBER}
  />
  <CarTextItem
    label="Орган внутрішніх справ"
    text={OVD}
  />
  <CarTextItem
    label="Дата крадіжки"
    text={THEFT_DATA}
  />
  <CarTextItem

```

```

        label="Регістраційний номер"
        text={VEHICLENUMBER}
      />
    </Card>
  </div>
)
export default memo(FoundedCars);

```

Б.5 Файл MainLayout.jsx

```

const { Header, Content, Footer } = Layout;
const MainLayout = ({ children }) => {
  const router = useRouter();
  return (
    <Layout
      style={{ height: '100%' }}
      className="layout"
    >
      <Header>
        <Menu
          theme="dark"
          mode="horizontal"
          items={[{
            key: '1',
            label: 'Пошук вкрадених',
            onClick: () => router.push('/')
          }]}
        />
      </Header>
      <Content style={{ padding: '0 50px', height: '100%' }}>
        <div className="site-layout-content">
          { children }
        </div>
      </Content>
      { /*<Footer style={{ textAlign: 'center' }}></Footer>*/ }
    </Layout>
  )
}
export default memo(MainLayout);

```

Б.6 Файл search.js

```
export default function handler({ query: { carNumber }}, res) {
  return res.send(getVehicleByNumber(carNumber));
}
```

Б.7 Файл [carNumber].js

```
const { Title, Text } = Typography
const WantedCar = ({ carInfo }) => {
  if (!carInfo) {
    return (
      <Typography
        style={{
          paddingTop: '15vh',
          textAlign: 'center'
        }}
      >
        <Title
          type="danger"
        >
          Ми не змогли знайти автомобіль
        </Title>
        <Text>
          Можливо, авто не знаходиться в розшуку, або Ви ввели неправильно номер
        </Text>
      </Typography>
    )
  }
  return (
    <FoundedCars
      {...carInfo}
    />
  )
}

export async function getServerSideProps({ params: { carNumber } }) {
  if (carNumber) {
    const { data : carInfo } = await searchApi({carNumber})
    return { props: { carInfo } }
  }
  return { props: { } }
```

```

}
export default WantedCar

```

Б.8 Файл _app.js

```

function MyApp({ Component, pageProps }) {
  return (
    <MainLayout>
      <Head>
        <title>Wanted Cars</title>
        <meta property="og:title" content="Wanted Cars" key="title" />
      </Head>
      <Component {...pageProps} />
    </MainLayout>
  )
}
export default MyApp

```

Б.9 Файл index.js

```

export default function Home() {
  return (
    <FindCar />
  )
}

```

Б.10 Файл car.js

```

const CarSchema = new Schema({
  "_id":String,
  "ID":String,
  "ODV":String,
  "BRAND":String,
  "COLOR":String,
  "VEHICLENUMBER":String,
  "BODYNUMBER":String,
  "CHASSISNUMBER":String,
  "ENGINENUMBER":String,
  "THEFT_DATA":String,
  "INSERT_DATE":String,

```

```
});
module.exports = mongoose.model('Cars', CarSchema);
```

Б.11 Файл loadData.js

```
const loadData = async () => {
  const { data: { result: { resources: [{ id } ] } } } = await
  axios.get(`${baseUrl}/package_show?id=${key}`)
  const { data: { result: { resource_revisions: [{ url } ] } } } = await
  axios.get(`${baseUrl}/resource_show?id=${id}`)
  const fStreamOut = fs.createWriteStream('../data/mvswantedtransport-1.json');
  https.get(url, response => response.pipe(fStreamOut));
}
export default loadData;
```

Б.12 Файл index.js

```
export const searchApi = ({ carNumber }) => axiosInstance.get('/search', {
  params: {
    carNumber
  }
})
```

Б.13 Файл global.css

```
html,
body {
  padding: 0;
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen,
  Ubuntu, Cantarell, Fira Sans, Droid Sans, Helvetica Neue, sans-serif;
}

#__next,
.main {
  height: 100%;
}

a {
  color: inherit;
  text-decoration: none;
```

```
}
```

Б.14 Файл package.json

```
{  
  "name": "car-finder",  
  "version": "0.1.0",  
  "private": true,  
  "scripts": {  
    "dev": "next dev --port 3005",  
    "build": "next build",  
    "start": "next start --port 3005",  
    "lint": "next lint"  
  },  
  "dependencies": {  
    "antd": "^4.20.4",  
    "axios": "^0.27.2",  
    "date-fns": "^2.28.0",  
    "formik": "^2.2.9",  
    "mongoose": "^6.3.3",  
    "next": "12.1.6",  
    "react": "18.1.0",  
    "react-dom": "18.1.0"  
  },  
  "devDependencies": {  
    "eslint": "8.15.0",  
    "eslint-config-next": "12.1.6"  
  }  
}
```

ДОДАТОК В
Слайди презентації

РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ ПОШУКУ АВТОМОБІЛІВ У РОЗШУКУ

Група: КНТ-129сп

Студент: Недошивко Віталій Сергійович

Керівник: к.т.н., доцент Казурова Аліна Євгенівна

Рисунок В.1 – Слайд № 1

Мета, об'єкт, предмет і завдання дослідження

Об'єкт дослідження – програмні засоби для пошуку транспортних засобів у розшуку.

Предмет дослідження – вебсайт для пошуку транспортних засобів у розшуку.

Мета роботи – створення вебзастосунку для швидкого пошуку транспортних засобів у розшуку.

Актуальність роботи полягає в тому, що останнім часом кількість викрадених автомобілів збільшується.

Рисунок В.2 – Слайд № 2

Огляд існуючих рішень

Вебсайт «УКРНОМЕР»

Параметри	АВТОНОМЕР	УКРНОМЕР
Інтерфейс	простий та зрозумілий	зрозумілий, але має багато даних
Швидкість пошуку	невелика	низька
Робота з іншими базами даних	у інших вкладках	у одному пошуку
Вартість	безкоштовний	частково безкоштовний

Рисунок В.3 – Слайд № 3

Вибір мови програмування для серверної частини

Параметр	PHP	Node.js	Java
Сфера застосування	Веброзробка (сервер)	Веброзробка (сервер)	Веброзробка (сервер), мобільні та desktop додатки
Наявність CMS	Багато	Небагато	Небагато
Швидкодія	Висока	Середня	Висока
Вартість	Безкоштовний	Безкоштовний	Безкоштовний
Паралельна робота	Багатопоточність	Асинхронність одного потоку	Багатопоточність
SSR	+	+	+

Рисунок В.4 – Слайд № 4

Вибір бібліотеки для клієнтської частини

Параметр	React.js	Vue.js	Angular.js	Svelte
Фреймворк / бібліотека / компілятор	бібліотека	фреймворк	фреймворк	компілятор
Віртуальний DOM	+	+	Incremental DOM	-
Шаблон проектування	V	MVVM	MVC	-
Поріг входження	низький	низький	великий	низький
Розмір готового застосунку	середній	маленький	великий	маленький

Рисунок В.5 – Слайд № 5

Вибір СКБД

Параметр	Тип бази даних	Легкість документації	Масштабованість	Легкість навчання
MySQL	SQL	середня	вертикальна, складна	середня
Maria DB	SQL	легка	вертикальна	середня
Oracle	Multi-model, SQL	легка	вертикальна	складна
PostgreSQL	Object-relational, SQL	середня	вертикальна	складна
MSSQL Server	T-SQL	легка	вертикальна, складна	складна
MongoDB	NoSQL, document-oriented	легка	горизонтальна	середня
Redis	NoSQL, key-value	легка	горизонтальна	середня
Elasticsearch	NoSQL, document-oriented	середня	горизонтальна	складна
Firebase	NoSQL, real-time database	легка	горизонтальна	середня

Рисунок В.6 – Слайд № 6

Схема бази даних



Рисунок В.7 – Слайд № 7

Загальна архітектура вебсайту

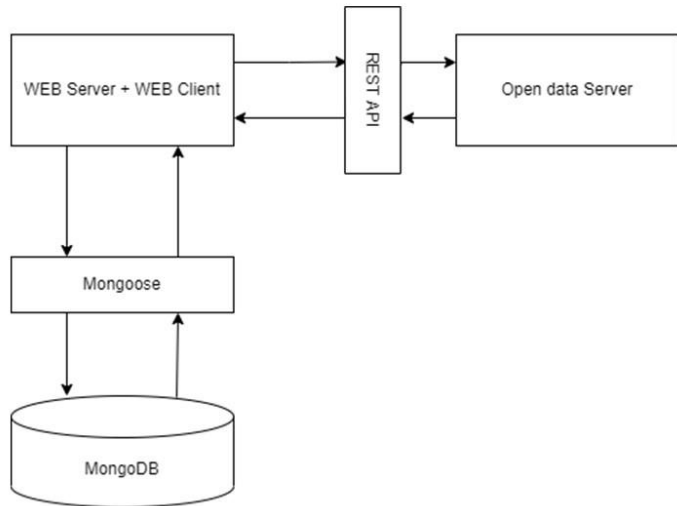


Рисунок В.8 – Слайд № 8

Скріншоти зробленої програми

Сторінка пошуку

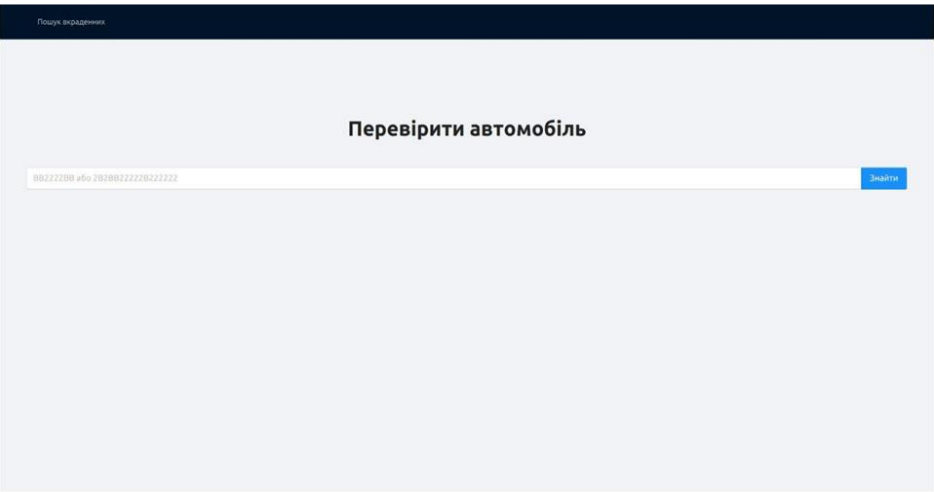


Рисунок В.9 – Слайд № 9

Скріншоти зробленої програми

Сторінка інформації про транспортний засіб

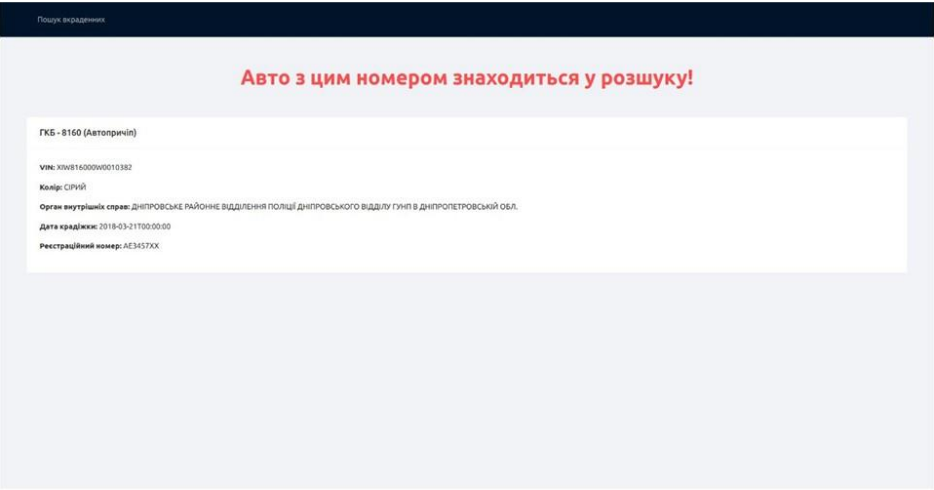
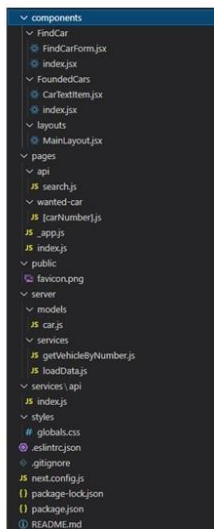


Рисунок В.10 – Слайд № 10

Файлова структура



- components – файли React компонентів;
- pages – файли сторінок;
- public – файли для публікації;
- server – файли серверної частини;
- services – файли сервісів, допоміжних функцій;
- styles – файли стилів;
- .eslinttrc.json – файл конфігурації eslint;
- .gitignore – файл конфігурації файлів та папок які не будуть зафіксовані у комміті;
- next.config.js – файл конфігурації Next.js;
- package.json – файл конфігурації залежності пакетів;

Рисунок В.11 – Слайд № 11

Висновки

Під час виконання дипломної роботи було розроблено програмне забезпечення для пошуку транспортних засобів у розшуку.

Було проведено аналіз предметної області та сучасних технологій, враховано можливі труднощі при розробці програмного забезпечення, та на їх основі вирішені наступні задачі:

- створення простого та зрозумілого користувацького інтерфейсу;
- реалізація швидкого пошуку.

Рисунок В.12 – Слайд № 12