

Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра обчислювальної техніки

### БАКАЛАВРСЬКА ДИПЛОМНА РОБОТА

на тему:

**Веб-додаток інформаційного обслуговування автопаркінгу**

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

Виконала студентка 4 курсу, групи 2КІ-186  
спеціальності 123 — Комп'ютерна інженерія

Кривенька В. О.

Керівник к.т.н., доц. каф. ОТ

Тарновський М. Г.

Опонент д.т.н., проф. каф. ЗІ

Лужецький В. А.

Допущено до захисту  
д.т.н., проф. Азаров О.Д.

" 20 " червня 2022 р.

ВНТУ 2022

Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра обчислювальної техніки  
Освітньо-кваліфікаційний рівень бакалавр  
Спеціальність 123 — «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

обчислювальної техніки

проф. Азаров О.Д.


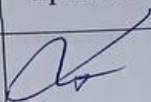
«08» 02 2022 р.

**ЗАВДАННЯ****НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ**

Кривенькій Вікторії Олегівні

1. Тема проекту «Веб-додаток інформаційного обслуговування автопаркінгу», керівник роботи к.т.н., доц. каф. ОТ Тарновський М. Г. затверджені наказом вищого навчального закладу від «24» березня 2022 року № 66
2. Строк подання студентом проекту 20.06.2022 р.
3. Вихідні дані до проекту: призначення — пошук та бронювання місця на паркінгу; основні підтримувані функції — зчитування QR-коду; використовувані сторонні ресурси — платіжна система.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ, аналіз предметної області. Розробка веб-додатку. Проектування веб-додатку.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): технічне завдання, головна сторінка веб-додатку.
6. Консультанти розділів роботи приведені в таблиці 1.

Таблиця 1 — Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Спеціальна частина	Тарновський М. Г, доцент кафедри ОТ	 24.03.22	 16.06.22

7. Дата видачі завдання « 24 » березня 2022 р.

8. Календарний план виконання БДР приведений в таблиці 2.

Таблиця 2 — Календарний план

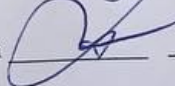
№ з/п	Назва етапів виконання комплексної бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Постановка задачі роботи	07.03.22	викон.
2	Аналіз предметної області	08.03-18.03.22	викон.
3	Аналіз сучасних підходів до побудови інформаційних систем паркінгу	21.03-31.03.22	викон.
4	Аналіз поняття веб-додатку	04.04-15.04.22	викон.
5	Аналіз принципів роботи веб-додатків	18.04-22.04.22	викон.
6	Проектування веб-додатку	25.04-06.05.22	викон.
7	Розробка веб-дизайну	09.05-13.05.22	викон.
8	Розробка функціоналу веб-додатку	16.05-20.05.22	викон.
9	Підготовка матеріалів та опис розробки інформаційної системи	23.05-26.05.22	викон.
10	Аналіз виконання роботи, висновки, додатки	27.05-31.05.22	викон.
11	Перевірка якості виконання бакалаврського проекту та усунення недоліків	01.06 - 08.06.22	викон.

Студентка



Кривенька В. О.

Керівник роботи



Тарновський М. Г.

## АНОТАЦІЯ

Кривенька В.О. Веб-додаток інформаційного обслуговування автопаркінгу. Бакалаврська кваліфікаційна робота зі спеціальності 123 — Комп'ютерна Інженерія, Вінниця: ВНТУ, 2022. Пояснювальна записка містить 69 сторінок, 16 рисунків та 20 посилань.

В даній бакалаврській дипломній роботі був розроблений додаток інформаційного обслуговування автопаркінгу. На основі здійсненого аналізу предметної області було проаналізовано сучасні підходи до побудови інформаційних систем паркінгу та наведено декілька аналогів. Відповідно до поставленої задачі був розроблений інтерфейс, з використанням HTML, CSS та JS. Таким чином було отримано зручну та швидкодіючу систему що спрощує водію пошуку вільних місць на парковках.

Ключові слова: автоматизоване паркування, додаток, інтелектуальна система, розумна паркова, QR-код.

## **ABSTRACT**

Kryvenka V. O. Car parking information service web application. Bachelor's degree in specialty 123 — Computer Engineering, Vinnytsia: VNTU, 2022. The explanatory note contains 69 pages, 16 figures and 20 references.

In this bachelor's thesis was developed an application for information services for car parking. Based on the analysis of the subject area, modern approaches to the construction of parking information systems were analyzed and several analogues were presented. In accordance with the task, an interface was developed using HTML, CSS and JS. This resulted in a convenient and fast system that simplifies the driver's search for parking spaces.

Keywords: automated parking, application, intelligent system, Smartparking, QR-code

## ЗМІСТ

<b>ВСТУП</b> .....	7
<b>1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ</b> .....	10
1.1 Аналіз сучасних підходів до побудови інформаційних систем паркінгу	10
1.2 Огляд аналогів .....	13
<b>2 РОЗРОБКА ВЕБ-ДОДАТКУ</b> .....	20
2.1 Поняття різниці між веб-додатком та веб-сайтом .....	20
2.2 Переваги та недоліки веб-додатку.....	22
2.2.1 Переваги створення веб-додатків.....	22
2.2.2 Недоліки створення веб-додатків.....	23
2.3 Принципи роботи веб-додатку .....	24
2.4 HTML та CSS .....	25
2.5 Java Script .....	33
<b>3 ПРОЕКТУВАННЯ ВЕБ-ДОДАТКУ</b> .....	36
3.1. Визначення принципів функціонування інформаційної системи паркінгу з підтримкою онлайн сервісів .....	36
3.2 Розробка веб-дизайну .....	39
3.3 Тестування веб-додатка.....	42
<b>ВИСНОВКИ</b> .....	49
<b>ПЕРЕЛІК ДЕЖЕРЕЛ ПОСИЛАННЯ</b> .....	50
<b>ДОДАТОК А</b> Технічне завдання .....	52
<b>ДОДАТОК Б</b> Лістинг HTML файлу.....	56
<b>ДОДАТОК В</b> Лістинг CSS файлу .....	61
<b>ДОДАТОК Г</b> Лістинг JS файл .....	67
<b>ДОДАТОК Д</b> Головна сторінка веб-додатку.....	68
<b>ДОДАТОК Е</b> Протокол перевірки навчальної (кваліфікаційної) роботи .....	69

					08-23.БДР.025.00.000 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Веб-додаток інформаційного обслуговування автопаркінгу. Пояснювальна записка	<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Розробила</i>		Кривенька В. О.						
<i>Керівник</i>		Тарновський М.					6	69
<i>Опонент</i>		Лужецький В.А.				ВНТУ, гр. 2КІ-186		
<i>Н.контр.</i>		Швець С. І.						
<i>Затвердж.</i>		Азаров О.Д						

## ВСТУП

Невід'ємною частиною міської інфраструктури є парковка. Ефективність використання паркувального простору актуальна як для офісних, житлових, так і для торгово-розважальних та адміністративних об'єктів. Останнім часом спостерігається диспропорція між темпами розвитку автомобілізації і темпами розвитку дорожньої мережі. Через це одним з центральних місць у транспортних проблемах міст стає дефіцит паркувального простору, що висуває завдання комплексного вирішення управління парковками [1], адже, згідно статистики, понад 20% трафіку у центрі міст складають водії, які шукають місце для паркування [2].

Сучасні системи управління автостоянки є чудовим рішенням, які дозволяють зробити автомобільні паркінги комфортними для користувачів і економічно ефективними для їх власників. Останнім часом вони стають незамінним засобом для водіїв, коли потрібно забезпечити платний в'їзд на контрольовану територію, підтримати гнучку систему тарифікації в залежності від часу паркування, підвищити заповнюваність паркінгу і зробити його комфортним. Автоматична фіксація в'їздів і виїздів, розрахунок вартості та прийом оплати значно збільшує пропускну здатність, скорочують експлуатаційні витрати, захищають від зловживань з боку користувачів і персоналу [3].

Автоматизована стоянка має неабиякі переваги, зокрема високий рівень безпеки, зручність і комфорт експлуатації економія площі, коштів та як на мене саме головне — часу. Згідно з дослідженнями, застосування системи автоматичного стоянки зменшує обсяг трафіку на 9%, викиди газу на 45%, кілометраж, пройдений автомобілем до парковки на 35% і час, що витрачається на парковку на 41% [4].

Завдяки інтелектуальній системі автоматичного паркування, користувачі можуть отримати можливість здійснювати пошук і бронювати вільні місця для свого автомобіля, а також перевіряти час закінчення дії паркувальної сесії. Одна з особливостей smart-парковки — обов'язкова

реєстрація, яка дозволить ідентифікувати транспортний засіб і його власника. SmartParking дозволить поліпшити систему оплати та бронювання, а також контролювати порушників [5].

Відповідно до цього, зручним у використанні для усіх водіїв стане веб-додаток паркінгу у місті. Водій знаходить найближчу парковку, отримує інформацію чи є ще вільні місця та бронює собі одне із цих місць. Натомість для того аби бути впевненим, що його місце більше ніхто не займе, йому на електронну пошту приходить лист із QR-кодом. При в'їзді у парковку, водій зобов'язаний піднести QR-код до зчитувача. Таким чином система зафіксує його в'їзд у парковку та позначить місце у веб-додатку як зайняте.

**Актуальність теми** дослідження є у розробці додатку для пошуку паркувальних місць, що підтверджується великою потребою цього сервісу у великих містах особливо зі збільшенням трафіку, який пропорційно збільшується зі зростанням кількості населення великих міст.

**Об'єктом досліджень** є процеси створення веб-додатку.

**Предметом досліджень** є методи проектування та технології створення веб-додатків.

**Метою роботи** є розробка веб-додатку для бронювання вільних паркувальних місць у місті.

Для досягнення поставленої мети необхідно виконати такі завдання:

- проаналізувати визначену предметну область;
- проаналізувати принцип роботи веб-додатків;
- розробити веб-додаток у вигляді веб-сторінки.
- розробити функціонал роботи додатку, що буде бронювати та займати вільні місця на парковці.

**Апробація** результатів роботи здійснена в доповіді на LI Науково-технічній конференції підрозділів Вінницького національного технічного університету.

Матеріали роботи доповідались та опубліковувались:



Кривенька В. О. Веб-додаток інформаційного обслуговування автопаркінгу/ В. О. Кривенька, М. Г. Тарновський // Тези доповіді. Лі регіональна науково-технічна конференція професорсько-викладацького складу, співробітників та студентів ВНТУ з участю працівників науково-дослідних організацій. Вінниця 2022 р.

Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2022/paper/view/14872/12567>.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Аналіз сучасних підходів до побудови інформаційних систем паркінгу

Автоматична паркова є досить актуальною на сьогоднішній день, так як з кожним днем кількість автомобілів у великих містах, та й не тільки, зростає з шаленою швидкістю. Зазвичай вони розташовані неподалік великих комплексів, спортзалів, торгових центрів, тощо. Усі водії звикли до простих парковок, але з розвитком нових технологій, досить популярними стають саме автоматизовані. Це обумовлено тим, що система автоматизації паркування дозволяє організувати зручний в'їзд або виїзд машин на парковку, що підвищує її пропускну здатність та робить привабливою для клієнтів. З іншого боку підвищується економічна ефективність для керуючої компанії за рахунок суттєвого зменшення імовірності зловживання з боку робітників.

Подібні парковки обов'язково мають такі елементи: стійки на в'їзді, виїзді, автоматичні шлагбауми, табло та елементи візуальної навігації, каси або термінали для оплати, систем відеоспостереження, тощо. Усі ці компоненти об'єднуються в єдину інформаційну систему, яка забезпечує їх спільне функціонування в єдиному комплексі [6].

Автоматизована паркова дозволяє вирішити певний ряд задач:

- автоматизувати процеси паркування автомобілів;
- автоматичним способом здійснити оплату послуг;
- надати звіт про кількість автомобілів на конкретній парковці, проведеному на ній часі, а також оплачені послуги [6].

Основними перевагами автоматизаційних парковок є економія грошей, адже впровадження автоматизованого паркування в багатьох випадках дозволяє повністю позбутися витрат на операторів, які на не автоматизованих паркінгах вручну управляють і регулюють процесами на паркування. Другою перевагою можна зазначити конкурентоспроможність, наявність можливості

вільно запаркуватися в безпосередній близькості від мети подорожі стає значною перевагою. Особливо в той час року коли клімат не схильний до піших прогулянок. Ефективність паркування — нераціональне використання паркувального простору. Чи не автоматизовані парковки заповнюють таксиста, офісні клерки з найближчих бізнес-центрів, мешканці прилеглих будинків тощо. Це заважає заповненню паркінгу цільовими відвідувачами. Наступною перевагою є контроль. За різними даними на платних, але не автоматизованих парковках, за участю операторів у процесах оплати, власники паркінгів не отримують до 35% виручки. Останньою з переваг це безпека. У кризових умовах зазвичай зростає кількість злочинів. Важливим завданням з яким вам допоможе впоратися система - це запобігання угону транспортних засобів [6].

Сучасні системи управління парковкою поділяються на дві основні групи: напівавтоматичні та автоматичні. Напівавтоматичні парковки бувають з однією стійкою або з двома. Напівавтоматична парковка з однією стійкою є найбільш бюджетним та поширеним варіантом. Вона передбачає постійну присутність персоналу (касира), який розміщується на місці виїзду з території парковки. Принцип роботи напівавтоматичної парковки з однією стійкою полягає в тому, що водій перед в'їзним шлагбаумом отримує в'їзний квиток зі штрих-кодом, стріла шлагбаума піднімається і система фіксує проїзд автомобіля. При виїзді водій віддає талон касиру, який за допомогою сканера штрих-кодів зчитує інформацію з талона після чого система автоматично розраховує вартість перебування даного автомобіля на парковці. Після отримання оплати, касир відкриває шлагбаум для виїзду[6].

Варіант з двома стійками використовується тоді, коли місце розташування касиру знаходиться не на виїзді із зони паркінгу. На відміну від попереднього варіанту після оплатити послуги паркінгу, водію надається виїзний талон, який прикладається до зчитувача на стійці при виїзному шлагбаумі. Автоматична парковка є повністю автоматизованим видом паркінгу. В'їзд, оплата та виїзд із зони паркінгу при такому варіанті

здійснюється водіями самостійно без допомоги обслуговуючого персоналу. В системі автоматичної парковки використовуються в'їзна та виїзна стійки, спеціальний термінал самообслуговування, який надає можливість оплатити послуги паркінгу самостійно. Така система в автоматичному режимі фіксує усі в'їзди, виїзди, а також формує фінансову звітність [7].

В інформаційних системах парковок можуть застосовуватися кілька видів ідентифікації. Найпростішим методом є ідентифікація за квитком зі штрих-кодом, який водій отримує при в'їзді. Штрих-код фіксується в базі даних. При виїзді квиток зчитується сканером штрих-кодів, на підставі чого розраховується сума для оплати.

Наступним варіантом є використання безконтактних RFID карток стандарту Mifare. На відміну від попереднього при такому підході з'являється можливість впровадження різних дисконтних програм на території комплексу. Найбільш просунутим варіантом є варіант, при якому ідентифікація автомобіля здійснюється за його номерним знаком. Для реалізації такого варіанту в'їзна та виїзна стійки обладнуються відеокамерами. У випадку, коли номерний знак зафіксувати не вдалося, водію автоматично видається квиток з штрих-кодом [7].

Усі з розглянутих варіантів побудови систем керування паркінгом не дозволяють водію заздалегідь дізнатись чи є вільні місця на паркувальному майданчику. Це створює передумови для того, щоб автомобілісти витрачали час на пошук вільного місця для паркування, проїзжаючи зайві кілометри. Наслідком цього є додаткове навантаження на дорожню мережу міста, що негативно впливає на безпеку руху та швидкість просування транспортних потоків.

Відповідно до цього розширення функціональних можливостей інформаційної системи обслуговування автопаркінгу, спрямоване на надання можливості водію заздалегідь визначати та бронювати місце для паркування є актуальним технічним завданням. Для його вирішення пропонується інтегрувати в інформаційну систему автопаркінгу веб-додаток. У результаті

водій отримує можливість переглянути наявність вільних місць за потрібною йому адресою, економлячи при цьому свій час та бензин. Якщо на парковці є вільне місце, водію надається можливість його забронювати на деякий час та згодом зайняти його [7].

Системи автоматичного паркування — це конструкції, в яких автомобілі розміщені вертикально, щоб обмежити простір. Конструкції цих унікальних систем допомагають транспортувати транспортні засоби від входу до паркувального місця без присутності водія. На додаток до системи автоматичного паркування, ці вертикальні паркувальні конструкції також відомі під кількома іншими назвами:

- роботизований гараж;
- система паркування автомобіля;
- автоматична парковка;
- автоматизована система зберігання та пошуку транспортних засобів;
- механічна стоянка [7].

Як це працює: водії підвозять свої автомобілі до входу системи автоматичного паркування, де всі пасажери виходять з автомобіля. Звідси автомобіль за допомогою механічних маневрів переміщується до вільного місця, де його автоматично припаркує або припаркує супровід[7].

## 1.2 Огляд аналогів

Одним з аналогів розглядуваної системи є автоматична система LOT PARKING.

LOT PARKING є сучасним рішенням, призначеним для організації автоматизованого контролю в'їзду та виїзду транспортних засобів та автоматизації розрахунків за користування паркувальними послугами.

Переваги автоматизованої системи паркування LOT PARKING:

- автоматизація системи оплати;
- фінансовий контроль та облік;

- моніторинг подій у режимі реального часу;
- безпека;
- зручність для клієнтів.

Автоматизована система паркування LOT PARKING вирішує такі завдання:

- автоматизація контролю в'їзду (виїзду) автомобілів на стоянку;
- автоматизація системи оплати на основі квитків зі штрих-кодом та безконтактних карт доступу;
- зручний контроль та аналіз фінансової діяльності паркувального комплексу;
- моніторинг подій системи LOT PARKING у режимі реального часу;
- комфортність користування послугами паркувального комплексу для клієнтів [8].

Система автоматизованого паркування LOT PARKING складається з в'їзної стійки, виїзної стійки, автомата оплати, сервера системи, автоматизованого робочого місця адміністратора, робочого місця касира та комплекту програмного забезпечення (рис. 1.1).

За бажанням в комплект постачання можуть входити в'їзний та виїзний шлагбауми.



Рисунок 1.1 — Проектування парковки за системою LOT PARKING

При в'їзді на паркування автомобіль потрапляє в зону дії першої в'їзної індукційної петлі, і система дозволяє в'їзд автомобіля на паркування. Водій повинен натиснути кнопку в'їзної стійки. Після того як водій забирає картку/білет зі штрих-кодом у в'їзної стійки, відкривається шлагбаум. Автомобіль перетинає першу та другу петлі, і в момент залишення другої петлі система подає команду на закриття шлагбауму. На екрані стійки з'являються підказки для водія. Якщо у водія виникли питання, він може скористатися голосовим зв'язком із адміністратором системи [8].

Оплата послуг паркування. Для оплати послуг паркування покупець може користуватися автоматично оплати або звернутися до касира. Клієнт повинен надати картку/квиток зі штрих-кодом, отриману на в'їзді. Система визначає час в'їзду та розраховує вартість послуги залежно від чинних тарифів. Клієнт зобов'язаний сплатити вартість паркування в автоматі оплати або у касира. В системі реєструється інформація про оплату та клієнту надається час для того, щоб залишити паркову [8].

При виїзді автомобіль потрапляє до зони першої виїзної індукційної петлі та запускається процедура виїзду. При виїзді клієнт вставляє картку у виїзну стійку. Система перевіряє факт оплати за пред'явленим документом (карта/квиток зі штрих-кодом). У разі успішного підтвердження оплати система відкриває виїзний шлагбаум. Клієнт отримує право виїзду і після залишення другої виїзної індукційної петлі система закриває шлагбаум. Система працює в режимі періодичного зв'язку з сервером, для нормального функціонування системи потрібен канал зв'язку між пристроями, що постійно працює [8].

Другим аналогом є автоматизація паркувальних місць на основі RFID. Ефективним рішенням для автоматизації паркування є RFID-система. Ринок не дуже багатий на подібні рішення, тому що знаходиться ще на стадії формування. Актуальність таких систем неухильно зростає, оскільки вони покликані вирішити ряд проблем, з якими стикаються сьогодні власники та користувачі паркувань [9].

Основні проблеми паркування:

- в'їзд/виїзд транспорту відбувається надто довго;
- відсутність контролю кількості вільних паркувальних місць;
- помилки та затримки через вплив людського фактора;
- відсутність точного обліку подій в'їзду та виїзду;
- RFID-обладнання на основі технології UHF дозволить оптимізувати контрольно-пропускні процеси [9].

Для цього застосовуються 4 основні компоненти автоматизації.

- UHF зчитувач дальньої дії;
- контролер системи контролю та управління доступом;
- UHF етикетці для ідентифікації автомобіля;
- програмне забезпечення [9].

Як працює система RFID на паркові? На автомобіль встановлюється RFID-мітка (наклейка під лобове скло або корпусна мітка під номерний знак; пластикова карта з чіпом дальньої дії) мітка потрапляє в поле видимості зчитувача (до 10 метрів) і зчитується, як результат унікальний код ідентифікатора потрапляє до контролера керованого спеціальним програмним забезпеченням (ПЗ). ПЗ дозволяє проводити адміністрування системи та отримувати аналітичні дані про пересування кожного автомобіля через КПП, а також оперативно управляти правами доступу для кожного користувача системи та багато іншого [9].

Впровадження системи автоматизованого в'їзду на паркінг дозволить скоротити кількість персоналу, який обслуговує паркінг, наприклад охоронців, а також мінімізувати вплив людського фактора на роботу системи.

Є можливість застосувати систему «Анти-дубль», яка запобігатиме доступу на паркування автомобілів з однаковими унікальними номерами міток. Повторний в'їзд автомобіля з цим унікальним номером мітки можливий лише після виїзду зчитування мітки на виїзному зчитувачі. Оскільки мітка нанесена на лобове скло має функцію саморуйнування, зафіксувати факт виїзду без фактичного проїзду на автомобілі неможливо [9].



### Переваги RFID-технології на паркова

- швидкісне безконтактне розпізнавання автомобіля (навіть при використанні спеціальної пластикової картки можна просто пред'явити її прямо з салону авто);
- відстеження час в'їзду/виїзду автомобілів;
- заборона проїзду транспорту без дозволу доступу та високий рівень захисту від несанкціонованого доступу;
- простота використання, надійність та безпека;
- легка установка та експлуатація практично без залучення людського ресурсу [9].

Третім аналогом є автоматизація паркування на основі технології UHF. Організація автоматичного паркінгу є невід'ємною частиною сучасного житлового комплексу. Вимоги користувачів паркування прості: забезпечення безпеки автомобіля, зручний та швидкий проїзд для економії особистого часу, наявність вільних місць. Класична система в'їзду на паркування вимагає докласти певних зусиль, які можуть бути вкрай незручними. Наприклад, висовуватись з вікна або чекати реакції охоронця [10].

На зміну високої вартості радіобрелків, необхідності часткої заміни елементів живлення, низькому рівню безпеки карток приходить автоматизована система в'їзду на основі технології ультра високочастотної ідентифікації (технології UHF) (рис 1.2).

З чого складається UHF-система:

- мітки ALN-9662, які є ідентифікатором автомобіля;
- зчитувачі [10].

Принцип дії системи полягає у встановленні спеціальної мітки, яка може бути інтегрована у різних формах: наклеєна під лобове скло, встановлена під номерний знак автомобіля або вбудована у звичайну пластикову картку.

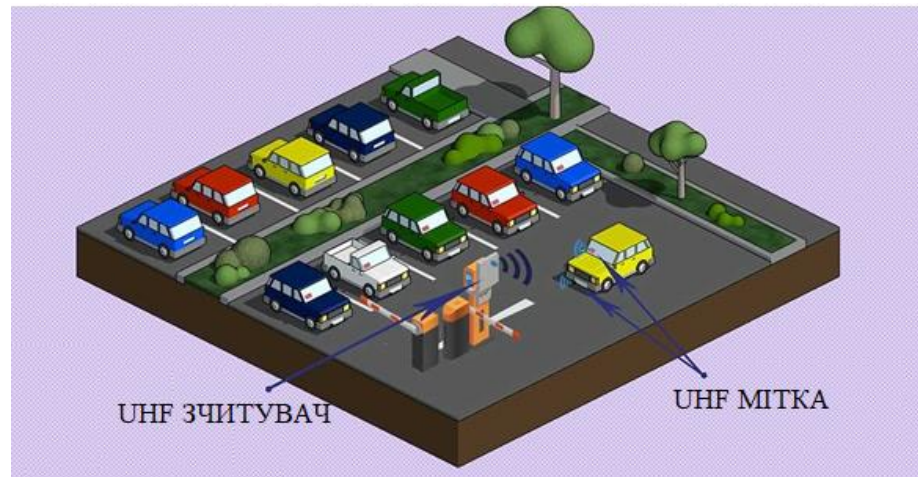


Рисунок 1.2 — Проекція парковки за системою на основі RFID

Установка мітки саме на автомобіль є найбільш раціональною, оскільки процес ідентифікації транспортного засобу у такому разі відбувається фактично без водія. Коли мітка потрапляє в поле видимості зчитувача, останній надсилає унікальний номер мітки для погодження на контролер, який керується спеціальним програмним забезпеченням. І вже контролер, проаналізувавши номер, приймає рішення про дозвіл чи заборону доступу [10].

Дальність зчитування міток при використанні цієї технології відбувається на відстані до 7 м (рис. 1.3).



Рисунок 1.3 — Мітка ALN-9662

У ПЗ можна відслідковувати час в'їзду/виїзду кожного автомобіля, призначати доступ для нього лише у певні зони паркінгу та багато інших функцій. Для випадків, коли у власника одного місця для паркування є кілька ідентифікаторів для доступу на це місце (наприклад, у вигляді пластикових карт), можна застосувати систему «Анти-дубль». Вона дозволить забороняти доступ на паркування міткам з однаковими номерами, поки цей номер не пройде через виїзний зчитувач. При цьому має бути зафіксовано реальний факт виїзду автомобіля через шлагбаум. Тобто пройти пішки з картою через виїзд і запустити ще одну машину за тим самим номером не вийде [10].

## 2 РОЗРОБКА ВЕБ-ДОДАТКУ

### 2.1 Поняття різниці між веб-додатком та веб-сайтом

Усім добре відомо, що з винайденням Інтернету його розвиток послужив деякою платформою для нового покоління людей для передачі інформації між собою та безперервного рівня доступу до будь-якої інформації. На початку всесвітньої павутини в основному складалася з веб-сайтів. В свою чергу веб-сайт складається з простої архітектури на основі HTML, CSS та JavaScript, і як правило служить лише для відображення певної інформації, він може містити зображення, текст, відео чи музику, але не може взаємодіяти з користувачем на вищому рівні. Наприклад, звичайний веб-сайт може містити деяку інформацію про серію продуктів у продуктовому магазині, проте він не має можливості здійснювати платежі через даний веб-сайт [11].

Веб-додаток — це програма, яка вбудована у веб-сторінку, в якому клієнтом виступає браузер, а сервером — веб-сервер. Зазвичай він написаний скрипковою мовою (PHP, Ruby), або ж написаний мовою високого рівня, які працюють на стороні веб-сервера та призначення переважно для створення інтерфейсу між користувачем та веб-сайтом. Або ж веб-додатки розробляються на основі кількох мов програмування, таких як Java, JavaScript, .Net, XML, AJAX та сервіси баз даних, такі як MySQL або Oracle [11].

Наприклад, веб-додатками є : веб-пошта, Gmail, Yahoo та AOL, онлайн-форми, , роздрібні продажі в Інтернеті, текстовий процесор, електронні таблиці, конвертація файлів, , редагування відео та фотографій, сканування файлів, Google Docs, Google Таблиці, Google Слайди, інтернет-сховища тощо.

У таблиці 1.1 наведена більш детальна характеристика, чим відрізняється веб-додаток від веб-сайту [11].

Таблиця 1.1 — Характеристика, різниці веб-додатку від веб-сайту

Параметр	Веб-додаток	Веб-сайт
Призначення	Розробляється переважно для взаємодії з користувачем	Розробляється для доступу до певної інформації, містить статті, без будь-якого можливого впливу на користувача
Авторизація	Веб-додатки частіше вимагають авторизації на від мінусу від веб-сайтів, так як вони пропонують користувачам більше можливостей, ніж веб-сайти. Більшість веб додатків вимагають авторизації ще на початку користування додатком, щоб користувач міг користуватись даною програмою	У деяких випадках авторизація використовується для того, щоб дати більше можливостей, які недоступні неавторизованим користувачем, проте не завжди, і не є обов'язковим. Хоча у деяких веб-сайтах є передбачений варіант, коли незареєстровані користувачі можуть лише переглядати статті, а зареєстровані - можуть залишити відгуки, ділитися статтями у соціальних мережах та багато іншого.
Взаємодія з користувачем	Користувач може проводити маніпуляції з даними, але з обмеженим доступом	Користувач може читати інформаційний контент, проте не може ніяк його змінювати
Складність	Веб-додаток має багато різних функцій	Веб-сайт відображає лише статичну сторінку, на якій зображена інформація

Отож, проаналізувавши таблицю 1.1 можна зробити висновок: що веб-додатком може бути деякий динамічний веб-сайт, який буде включати в себе взаємодію з користувачем, мати багато функцій, вимагати обов'язкову авторизацію та надавати можливість користувачу маніпулювати даними, проте в обмеженому доступі.

Веб-програми можуть використовувати веб-браузери та веб-технології для виконання певних функцій в Інтернеті. Зовнішній інтерфейс веб-програми розроблено такими мовами програмування: HTML, CSS, Javascript, які підтримуються будь-яким браузером (Opera, Chrome, Mozilla, Yandex). А для написання серверної частини (бекенда) можна використовувати будь-яку іншу мову програмування або фреймворк, Python, PHP, Ruby, Java [11].

## 2.2 Переваги та недоліки веб-додатків

### 2.2.1 Переваги створення веб-додатків

До переваг веб-додатків можна віднести їхнє застосування на будь-якій ОС. Вони підтримують як Linux, Windows та і Mac, оскільки ці операційні системи підтримують сучасні браузери. Зазвичай веб-додаток набагато простіше програмувати, оскільки він не включає багато роботи з елементами ПК (процесор, відеокарта, тощо). Веб-додатки не потребують безпосереднього встановлення на пристрій. Ці програми можуть працювати у власному веб-браузері пристрою через просту URL-адресу. Їх не потрібно завантажувати та встановлювати з магазинів додатків, таких як Google Play або Apple Store App. Завдяки цьому розробники економлять гроші, оскільки пряме посилання через веб-додаток є безкоштовним. Також користувачі можуть відкривати веб-сайти. Це означає, що вони не потребують оновлення, як це роблять звичайні програми [12].

Ще одною перевагою є захищеність даних. Веб-додаток легко підтримує низькі характеристики ПК або ж смартфонів. Це усуває проблему сумісності, так як користувач може отримати доступ до тієї ж версії. За

допомогою веб-програми працівники можуть працювати з будь-якого місця, користуючись доступом до Інтернету.

Також перевагами розробки веб-додатків включають у себе можливість збільшення зберігання веб-програм, оскільки існує хмарне сховище. Не потрібно ніякої установки, оскільки всі системи матимуть браузер; таким чином усувається обмеження простору. Веб-додаток розширює товар і послуги замовника, збираючи існуючі відгуки клієнтів, та використовуючи веб-програми, ви контактуєте з будь-ким і будь-де в світі [12].

### 2.2.2 Недоліки створення веб-додатків

Для того щоб користуватись веб-додатком, користувач може використовувати як телефон, так і планшет або ж ноутбук чи ПК. Але кожен з цих пристроїв має своє розширення екрану. І ось постає проблема, у правильності відображення веб-додатку. Він повинен бути адаптивний, тобто щоб він відображався незалежно від операційної системи пристрою, для того, що користувачу було комфортніше та зрозуміліше. Якщо це не адаптивний веб-сайт, у вас можуть виникнути проблеми під час його відображення на iOS, Android або Windows Phone.

Також веб-додаток повинен завжди мати доступ до інтернету, в іншому випадку, його завантаження не відбудеться, ви не зможете переглядати веб-сайт, і даний додаток не матиме ніякої користі. На жаль, у нашій країні не всюди є інтернет, або ж не всі люди можуть дозволити собі гаджет зі встроєним мобільним інтернетом, тож це дуже ускладнює роботу з веб-додатками, наприклад під час подорожі за містом, де 100% не буде ловити інтернет. Оскільки кожна сторінка затримується через встановлення оновлення з'єднання HTTP, то насамперед виконується запит на сервер, потім надсилаються HTTP-повідомлення у відповідь на мережу та перезавантаження сторінки відбувається за допомогою браузера, що, у свою чергу, перериває та сповільнює роботу [12].

### 2.3 Принципи роботи веб-додатку

Для розуміння принципів роботи веб-додатків необхідно знати що таке веб-сервер для обробки запитів користувачів, та як працює сервер додатків для виконання необхідних завдань з базою даних для зберігання інформації.

Розглянемо детальніше як це працює. Першим кроком користувач надсилає запит на веб-сервер через мережу Інтернет. Він використовує, як правило, веб-браузер. Далі веб-сервер надсилає запит користувача на конкретний сервер веб-додатків. Сервер у свою чергу виконує задану функцію — запит до бази даних та обробка запитуваних даних. Далі сервер пересилає дані користувача на веб-сервер, і вкінці виводить на екран потрібну інформацію користувача [13].

Отож, по самій структурі додатку, вся програмна логіка сконцентрована в одному місці, тобто на сервері, а користувацький інтерфейс доступний будь-якій людині у вигляді невеликої програми, яка існує з самого моменту зародження HTTP-мережі. Для появи веб-додатків не вистачало тільки таких важливих технологій, як JavaScript і DOM [13].

JavaScript — це сценарій клієнтської мови, яка використовується для створення веб-сторінок. Він використовується, коли веб-сторінку потрібно зробити динамічною та додати спеціальні ефекти на такі сторінки, як перекидання, розгортання та багато видів графіки. На додаток до валідацій, він підтримує зовнішні програми, такі як PDF-документи, запущені віджети, підтримує флеш-програми тощо. Він також може завантажувати вміст у документ, коли користувач цього вимагає, навіть не перезавантажуючи всю сторінку [13].

Завдяки технологіям браузерів та персональних комп'ютерів вдосконалено настільки, що для створення надійних веб-додатків потрібна мова. Коли користувач досліджує карту на Картах Google, користувачеві потрібно просто натиснути та перетягнути мишу. Весь детальний вид видно лише одним клацанням миші. Це можливо завдяки JavaScript. Він взаємодіє з



браузером, не надсилаючи повідомлення туди-сюди на сервери. Динамічна складова веб-сторінок завжди була слабкою частиною при створенні веб-додатків, і мабуть це єдина причина, чому веб-додатки зовсім нещодавно стали набувати популярності. Для того щоб користувач міг почати роботу з веб-додатком, йому потрібна тільки одна програма і це браузер, набравши URL вже можна приступати до роботи. Як правило, якщо користувач встановлює на своїй машині програму, йому доводиться брати на себе роль його адміністратора. Йому треба встановлювати програму, запускати, налаштовувати, лагодити, вирішувати проблеми, що виникають [13].

У випадку ж з веб-додатком, так як він розташовується на сервері, користувачеві немає необхідності турбуватися про це, на відміну від звичайного додатку. У ролі адміністратора виступає розробник додатку.

В веб-застосунках існує дві програми, які працюють одночасно: це код, який знаходиться на сервері і відповідає на HTTP запити; та код, який знаходиться в браузері і реагує на введення даних користувачем.

Код сервера невидимий для користувача, він може відповідати лише HTTP-запиту до певної URL-адреси і не може відповідати будь-якому входу користувача. Клієнтський код розбирається браузером користувача. На відміну від серверного він може реагувати на користувацькі введення, доступний для перегляду та редагування користувачеві у повному обсязі [13].

Отже, розробка веб-застосунків надає широкі можливості зі створення для компаній багатофункціональних онлайн-інструментів для оптимізації або вирішення різних бізнес-задач.

## 2.4 HTML та CSS

HTML означає мову розмітки гіпертексту. Він складається з ключових слів і команд, які веб-дизайнери використовують для створення веб-сайтів. Гіпертекст – це текст із посиланнями, на які читачі можуть просто натиснути, щоб перейти на іншу сторінку чи іншу частину сторінки. Тим часом мова розмітки використовує теги або звичайний текст зі спеціальними позначками

для визначення розділів сторінки, таких як верхні та нижні колонтитули, та інших елементів, включаючи таблиці та зображення [14].

HTML вважається одним із трьох основних інструментів у створенні веб-сторінки: HTML забезпечує структуру або спосіб відображення тексту, зображень тощо на веб-сайті. CSS (каскадні таблиці стилів) встановлює візуальні властивості цих елементів, такі як кольори, формат і макет. Тим часом Javascript змушує ці елементи вести себе певним чином залежно від дій користувача. Наприклад, розмір шрифту тексту може збільшуватися, коли користувач наводить курсор миші або натискає кнопку на сторінці [14].

Щоб написати HTML, вам знадобиться текстовий редактор, наприклад Блокнот або Sublime Text. Редактори HTML переконуються, що ваше кодування є чистим і функціональним. Вони допомагають зменшити кількість помилок, автоматично вставляючи теги (автозавершення) та інші поширені елементи шляхом налагодження.

HTML — мова за замовчуванням для веб-сайтів і веб-документів. Це допомагає браузеру зрозуміти структуру та стиль документа або файлів для перегляду через Інтернет. Це дозволяє на ваших веб-сторінках розміщувати аудіо, відео, електронні таблиці та інші програми. Це також полегшує навігацію по веб-сторінках або між веб-сайтами за допомогою гіпертексту [14].

Крім того, розробники веб-сайтів можуть використовувати HTML для розробки форм для замовлення товарів, бронювання або пошуку інформації. Таким чином, HTML є основним будівельним блоком для створення вашого бренду та ведення сайту електронної комерції або бізнесу на основі підписки.

Веб-сторінки у всесвітній павутинні переглядають не тільки люди. Для людини немає особливих труднощів розібратися в контенті сторінки, тобто вона одразу зрозуміє, де знаходиться меню (навігація), заголовки, основний зміст, додатковий контент, контактна інформація і т.д. Але, окрім людини, проаналізувати контент веб-сторінки намагаються й інші системи: пошукові роботи, програми, що читають текст з екрана тощо. Для них дуже важливо,

те як створено сайт, тобто з яких елементів він складається і як ці елементи взаємопов'язані між собою. Отже, однією з основних завдань, яку необхідно виконати веб-розробнику - це вкласти сенс у створювані веб-сторінки, тобто. зробити їх якомога зрозумілішими для пошукових та інших систем [14].

Процес вкладення сенсу у вміст веб-сторінки за допомогою елементів HTML 5 і їх взаємного розташування називається семантикою. Чому так важливо зробити сайт семантично вірним? Так, тому що чим зрозумілішим стане контент сайту для пошукових роботів, тим його простіше буде знайти користувачам [14].

Таким чином, семантика, це дуже важлива частина сучасного веб, це його серце. При цьому чим більш семантично вірно побудований сайт, тим він є більш ефективним.

При створенні HTML-документа необхідно дуже ретельно підходити до вибору та розташування елементів. Необхідно знати який семантичний сенс стоїть за кожним із використовуваних елементів і як він поєднується з іншими елементами. І вже потім переходити до розмітки деякого контенту [14].

Основні напрямки, над якими необхідно працювати автору з HTML 5 документом, щоб зробити його семантично вірним:

- використовувати під час розмітки веб-сторінки нові семантичні елементи HTML 5.
- використовувати нові правила HTML 5 створення змісту документа, тобто. покращувати його структуру.
- застосовувати розробки нових моделей контенту, які описують як і де можна використовувати ті чи інші елементи [14].

Кожен HTML-документ, який відповідає специфікації HTML будь-якої версії, повинен починатися з рядка оголошення версії HTML `<!DOCTYPE ...>`, який зазвичай виглядає приблизно так: `<!DOCTYPE html>`. У даному випадку ми говоримо браузеру, що HTML відповідає міжнародній специфікації версії 5 [15].

Не треба намагатися запам'ятати цей рядок. Головне - необхідно знати, що він необхідний. Адже саме він допомагає зробити сайт, який буде однаково виглядати в усіх браузерах (як правило це Mozilla Firefox, Google Chrome, Opera, Internet Explorer) [15].

Далі позначається початок і кінець документа тегами `<html>` і `</html>`, відповідно. Всередині цих тегів повинні знаходитися теги голови (`<head>` `</head>`) і тіла документу (`<body>` `</body>`) (рис 2.1).

```
<!DOCTYPE HTML>
<html>
<head>
<title>Структура HTML документа</title>
</head>
<body>
<h1>Hello HTML!</h1>
</body>
</html>
```

Рисунок 2.1 — Структура HTML документа

Кодування HTML-сторінки потрібно вказувати для того, щоб веб-браузер міг правильно відображати текст на сторінці. Якщо браузер неправильно «вгадає» кодування, то замість тексту будуть відображатися ієрогліфи. Щоб повідомити браузеру кодування HTML-сторінки, необхідно всередині тега `<head>` використовувати тег: `<meta charset = "ім'я кодування">`. Найпоширеніше сучасне кодування - utf-8. Використовуйте його в усіх своїх проектах. Для кирилиці в Windows charset часто задавали як windows-1251. Але зараз це вважається поганою практикою [15].

До речі, в HTML-редакторі можна побачити такі фрагменти коду:

```
<!-- Деякий текст -->
```

Їх називають «коментарі», і браузер не буде відображати їх на сторінці.

Зазвичай коментарі використовуються в наступних випадках:

- для коментування коду;

— для тимчасового відключення коду.

Коментарі можна використовувати в будь-якому місці сторінки, крім тегу <title> – всередині якого вони не працюють.

Основними елементами форматування тексту в html є теги:

- p використовується для розбивання тексту на параграфи;
- h1, h2, .... h6 застосовуються для створення заголовків 1,2 ... 6 рівнів, відповідно;
- br використовується для переносу рядка;
- div, span використовуються для виділення частини документа певним способом.

Наведені теги — це далеко не повний перелік елементів, за допомогою яких форматується текст веб-сторінки. Але їх може бути достатньо для її створення.

Отже HTML це стандартизована мова розмітки документів для перегляду веб-сторінок у браузері, за допомогою якої можна створити веб-сайт. Вивчення основ HTML корисно як для майбутнього програміста, так і для всіх, хто хоче навчитися створювати веб-сторінки, які він чи вона щодня читає на комп'ютері чи смартфоні [15].

CSS ( Cascading Style Sheets або ж Каскадні таблиці стилів) — це спеціальна мова для стилізації веб-сторінок, що використовується для обробки зовнішнього вигляду веб-сторінки. Використовуючи CSS, можна змінювати колір тексту, його розмір та шрифт. Задавати колір фону, робити елементи прозорими, або ж взагалі сховати їх. Змінювати інтервал між літерами чи блоками, позиціонувати елементи, задавати ширину та висоту, робити сторінку адаптивною та багато чого іншого. CSS насправді легко вивчити та зрозуміти. Він забезпечує потужний контроль над поданням HTML-документа. Найчастіше CSS поєднується з мовами розмітки HTML або XHTML [16].

Наведемо декілька переваг CSS. CSS економить час — ви можете написати код один раз, а потім повторно використовувати той самий файл на

кількох сторінках HTML. За допомогою CSS можна визначити стиль для кожного елемента HTML і застосувати його до будь-якої кількості веб-сторінок. Сторінки швидше завантажуються, якщо ви використовуєте CSS, вам не потрібно щоразу писати атрибути тегу HTML. Просто напишіть одне правило CSS для тегу та застосуйте його до всіх входів цього тегу. Таким чином, менше коду буде означати швидше завантаження [16].

Просте обслуговування — щоб внести глобальні зміни, просто змініть стиль, і всі елементи на всіх веб-сторінках оновляться автоматично. CSS має набагато ширший набір атрибутів, ніж HTML, тому ви можете надати набагато кращий вигляд своїй HTML-сторінці порівняно з атрибутами HTML. Сумісність із багатьма пристроями — таблиці стилів дозволяють оптимізувати вміст для різних типів пристроїв. Використовуючи той самий документ HTML, можна представити різні версії веб-сайту для портативних пристроїв, таких як КПК і мобільні телефони, або для друку [16].

Структура стилю складається з трьох частин:

— селектор — це тег HTML, до якого буде застосований стиль, це може бути будь-який тег, наприклад `<h1>` або `<table>` тощо;

— властивість — це тип атрибута тегу HTML, простіше кажучи, усі атрибути HTML перетворюються у властивості CSS, вони можуть бути кольоровими, бордюрними тощо;

— значення — призначаються властивостям, наприклад, властивість кольору може мати значення червоного або `#F1F1F1` тощо.

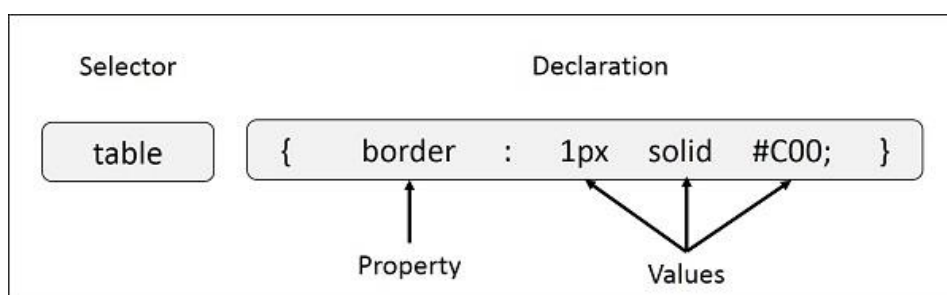


Рисунок 2.2 — Структура CSS

Є чотири способи пов'язувати стилі з вашим HTML-документом. Найпоширенішими методами є вбудований CSS і зовнішній CSS.

Вбудований CSS — елемент `<style>`. Ви можете помістити свій код CSS в HTML-документ за допомогою елемента `<style>`. Цей тег розміщується всередині тегів `<head>...</head>`. Правила, визначені за допомогою цього синтаксису, будуть застосовані до всіх елементів, доступних у документі [17].

Зовнішній CSS — елемент `<link>`. Елемент `<link>` можна використовувати для включення зовнішнього файлу таблиці стилів у ваш HTML-документ. Зовнішня таблиця стилів — це окремий текстовий файл із розширенням `.css`. Ви визначаєте всі правила стилю в цьому текстовому файлі, а потім можете включити цей файл у будь-який HTML-документ за допомогою елемента `<link>` [17].

CSS підтримує ряд вимірювань, включаючи абсолютні одиниці, такі як дюйми, сантиметри, точки тощо, а також відносні міри, такі як відсотки та одиниці `em`.

Так звані абсолютні одиниці (см, мм, дюйми, пт і пк) означають в CSS те саме, що й скрізь, але лише якщо ваш пристрій виведення має досить високу роздільну здатність. На лазерному принтері 1 см повинен бути рівно 1 сантиметр. Але на пристроях з низькою роздільною здатністю, таких як екрани комп'ютерів, CSS цього не вимагає. І дійсно, результат має тенденцію відрізнятися від одного пристрою до іншого та від однієї реалізації CSS до іншої. Ці одиниці краще зарезервувати для пристроїв з високою роздільною здатністю, зокрема для друкованої продукції. На екранах комп'ютерів і портативних пристроях ви, ймовірно, не отримаєте того, що очікуєте [17].

Є ще одна причина уникати абсолютних одиниць для інших цілей, ніж друк: ви дивитеся на різні екрани з різної відстані. 1 см на екрані робочого столу виглядає малим. Але те ж саме на мобільному телефоні прямо перед очима виглядає великим. Натомість краще використовувати відносні одиниці, наприклад `em` [17].

Одиниці `em` і `ex` залежать від шрифту і можуть відрізнятися для кожного елемента в документі. `Em` — це просто розмір шрифту. Таким чином, в елементі зі шрифтом 2 дюйми `1em` означає 2 дюйми. Визначення розмірів, таких як поля та відступи, в `em` означає, що вони пов'язані з розміром шрифту, і якщо користувач має великий шрифт (наприклад, на великому екрані) або дрібний шрифт (наприклад, на портативному пристрої), розміри будуть відповідні. Такі оголошення, як текстовий відступ: `1.5em` і `margin: 1em`, надзвичайно поширені в CSS [17].

`Ex` використовується рідко. Його мета полягає в вираженні розмірів, які повинні бути пов'язані з висотою шрифту по осі `x`. Висота `x` — це, приблизно, висота малих літер, таких як `a`, `s`, `m` або `o`. Шрифти, які мають однаковий розмір (`i`, отже, однаковий `em`), можуть сильно відрізнятися за розміром своїх малих літер, і коли важливо, щоб деяке зображення, наприклад, відповідало висоті `x`, доступний `ex` unit [17].

Одиниця `px` — це магічна одиниця CSS. Він не пов'язаний із поточним шрифтом і зазвичай не пов'язаний із фізичними сантиметрами чи дюймами. Одиниця пікселів визначається як маленька, але видима, і така, що горизонтальна лінія шириною 1 пікс може відобразитися з гострими краями (без згладжування). Те, що є різким, маленьким і видимим, залежить від пристрою та способу його використання: ви тримаєте його біля очей, як мобільний телефон, на відстані витягнутої руки, як монітор комп'ютера, чи десь посередині, як електронний читач книг? Таким чином, `px` визначається не як постійна довжина, а як щось, що залежить від типу пристрою та його типового використання [17].

Щоб зрозуміти звідки з'явилась одиниця `px` (піксель), уявіть собі, CRT монітор комп'ютера з 1990-х років: найменша точка, яку він міг відобразити вимірюється 1/100-тою дюйма (0.25мм) або трохи більше. Одиниця `px` отримала свою назву від тих екранних пікселів [18].

Щоб ще більше спростити, потрібно зробили правила стилів, які залежать тільки від розміру шрифту за замовчуванням, нова одиниця



знаходиться у стадії розробки: `rem. rem` (“root em”) (кореневий em) це розмір шрифту кореневого елемента документа. На відміну від `em`, яка може бути різною для кожного елемента, `rem` є постійною для всього документа. Інші одиниці, що розробляються, дозволяють вказати розміри відносно вікна читача. Це — `vw` і `vh`. `vw` є 1/100-ою ширини вікна і `vh` є 1/100-ою висоти вікна [18].

## 2.5 Java Script

JavaScript — це мова програмування, яка використовується як на стороні клієнта, так і на стороні сервера, та дозволяє розробляти веб-сторінки інтерактивними. Якщо HTML і CSS є мовами, які надають структуру та стиль веб-сторінкам, то JavaScript надає веб-сторінкам інтерактивні елементи, які залучають користувача. Поширені приклади JavaScript, які ви можете використовувати щодня, включають вікно пошуку на Amazon, відео з підсумком новин, вбудоване в The New York Times, або оновлення вашої стрічки Twitter [19].

Підключення JavaScript покращує роботу веб-сторінки, перетворюючи її зі статичної сторінки в інтерактивну, тобто JavaScript додає поведінку веб-сторінкам.

JavaScript використовується не тільки для веб-додатків і веб-браузерів, але й також за межами Інтернету в програмному забезпеченні, серверах та вбудованих апаратних елементах керування [19].

JavaScript дозволяє користувачам взаємодіяти з веб-сторінками. Практично немає обмежень для того, що клієнт може робити за допомогою JavaScript на веб-сторінці, але наведемо декілька прикладів:

- показати або приховати додаткову інформацію одним натисканням кнопки;
- змінити колір кнопки, коли миша наведе на неї курсор;
- прокрутити карусель із зображеннями на домашній сторінці;
- збільшення або зменшення масштабу зображення;

- відображення таймера або зворотного відліку на веб-сайті;
- відтворення аудіо та відео на веб-сторінці;
- відображення анімації;
- використання спадного меню гамбургерів [19].

Розробники можуть використовувати різні фреймворки JavaScript для розробки та створення веб та мобільних додатків. Фреймворки JavaScript — це колекції бібліотек коду JavaScript, які надають розробникам попередньо написаний код для використання для рутинних функцій і завдань програмування — буквально фреймворк для створення веб-сайтів або веб-додатків. Популярні інтерфейсні фреймворки JavaScript включають React, React Native, Angular і Vue. Багато компаній використовують Node.js, середовище виконання JavaScript, побудоване на движку JavaScript V8 Google Chrome. Кілька відомих прикладів включають PayPal, LinkedIn, Netflix та Uber [19].

Крім веб-сайтів і програм, розробники також можуть використовувати JavaScript для створення простих веб-серверів і розвитку внутрішньої інфраструктури за допомогою Node.js[19].

Структурно JavaScript можна представити у вигляді об'єднання трьох частин, що чітко відрізняються одна від однієї :

- ядро (ECMAScript);
- об'єктна модель браузера (об'єктна модель браузера або BOM);
- об'єктна модель документа (Document Object Model або DOM).

Об'єктну модель документа іноді розглядають як окрему від сутності JavaScript, що узгоджується з визначенням DOM як незалежного від мови інтерфейсу документа [19].

ECMASc не є браузером мовою і насправді в новій не подається методи введення та виведення інформації. Це скоріше основа для побудови скриптових мов. Специфікація ECMAScript описує типи даних, інструкції, ключові та зарезервовані слова, оператори, об'єкти, регулярні вирази, не обмежуючи авторів похідних мов від розширення їх новими складовими.

Об'єктна модель браузера - браузероспецифічна частина мови, яка з'являється про шарком між ядром і об'єктною моделлю документа. Основне призначення об'єктної моделі браузера - керування вікнами браузера та забезпечення їх взаємодії. Кожне з вікон браузера представляється об'єктом window, центральним об'єктом BOM [19].

Об'єктна модель документа - інтерфейс програмування додатків для HTML і XML-документів. згідно з документом DOM можна поставити у відповідність деревооб'єктів, які мають ряд властивостей, які можуть виконувати з ним різні маніпуляції: отримання вузлів, зміни вузлів, зміни зв'язку між вузлами, видалення вузлів [19].

### 3 ПРОЕКТУВАННЯ ВЕБ-ДОДАТКУ

3.1. Визначення принципів функціонування інформаційної системи паркінгу з підтримкою онлайн сервісів

Існують два види паркувальних систем: автоматичний та напівавтоматичний паркінг. Напівавтоматичний відрізняється лише тим, що на парковці присутній персонал, який буде обслуговувати водія. Це може бути охоронець, паркувальник, касир. Але так як наша задача позбутися будь-якого персоналу та перевести все на автоматизовану роботу, тож нам підходить перший вид паркувальної системи, розглянемо його принцип роботи.

Принципом роботи автоматизованих парковок є такий підхід: водій під час поїздки вирішив припаркуватись. Першим кроком є бронювання вільного місця на парковці. Він заходить на веб-додаток, обирає найближчу до нього парковку та якщо є вільне місце, бронює його на деякий час. За це час він повинен доїхати до вибраної ним адреси. Після натискання кнопки “забронювати”, йому на пошту приходить лист з QR-кодом. При в'їзді на парковку водій зупиняється біля пристрою читання QR-кодів та показує його зчитувачу. Система перевіряє чи дійсний та коректний QR-код і відкриває шлагбаум. Якщо несподівано, з якоїсь причини, автомобіль зупиниться під шлагбаумом, шлагбаум не вдарить по машині, так як індукційні петлі відчують перебування автомобіля під стрілою шлагбауму, а також вдарити не дасть другий ступінь безпеки у вигляді ІК давача У випадку, якщо QR-код не дійсний шлагбаум не підніметься.

Водій займає обране місце таке як і у додатку. Весь час поки автомобіль перебуває на стоянці, за ним спостерігає кілька відеокамер. При виїзді водій повинен знову показати свій QR-код системі, та система зчитуючи його вдруге розуміє, що водій хоче виїхати і відкриває шлагбаум, після цього з карти водія автоматично списуються кошти. Картку водій повинен прив'язати до додатку під час реєстрації.

Розглянемо варіант, якщо ж у водія буде недостатньо коштів, аби сплатити за користування парковкою. Водій зможе в'їхати та виїхати без проблем, так як система буде запам'ятовувати та накопичувати борг. І коли водій все ж з часом поповне свій банківський рахунок, необхідні кошти одразу ж будуть списані. Якщо борг буде перевищувати задану суму, водію прийде нагадування та буде включатись пеня.

Автоматизована система паркування захищає від зловживань, а саме - неможливо виїхати із території парковки, не маючи при собі необхідний QR-код, який впустив водія на парковку. На кожному куточку території, за всім спостерігають десятки камер з різних ракурсів, що дозволить спостерігати за тим як в'їхав автомобіль та виїхав. Камери зафіксують час перебування автомобіля на парковці.

Використання системи управління паркуванням має багато переваг, але водночас й містить деякі недоліки. Ці недоліки є причиною того, що люди не задумуються про використання систем керування паркуванням. Розглянемо деякі з них.

- висока вартість будівництва або монтажу;
- регулярне технічне обслуговування;
- невизначеність конструкції;
- запровадження нових та невідомих систем.

Вартість надійної, працюючої системи керування паркуванням зазвичай висока. Це пов'язано з різними компонентами, які беруть участь у забезпеченні роботи системи. Такі компоненти, як статистичні функції, автоматичне оформлення квитків, статистичні звіти та багато інших, роблять це все дорогим. Деякі організації не можуть дозволити собі це [20].

Система є автоматизованою, однак компанія все ще потребує регулярного обслуговування. Це потрібно для того, щоб переконатися, що система працює ідеально і що нічого не пішло не так. Технічне обслуговування бажано проводити раз на місяць.

Багато людей не звикли до системи управління паркуванням. Як наслідок, їм може бути важко користуватися, що спричиняє додаткові ускладнення під час паркування.

Розглянемо варіанти вирішення даних проблем. Проблему вищої вартості будівництва та монтажу можна вирішити за допомогою обговорення з виробниками про саму ціну. Але потрібно розуміти, що якісне обладнання є доволі дорогим. Тож на цьому точно заощаджувати не варто [20].

У проблемі експлуатації для багатьох нових користувачів, які не знайомі з системою керування паркуванням, спочатку їх використання може бути трохи незрозумілим. Цю проблему, звичайно, можна подолати за допомогою знайомства і невеликої освіти. Якщо вони задають питання про процедури експлуатації та дізнаються про переваги, які дає використання системи, вони з часом опанують її [20].

Щоб система працювала бездоганно, зазвичай необхідно проводити планове технічне обслуговування. Це робиться для того, щоб система паркування була в належному стані та щоб усі операції відбувалися відповідно. З постачальником або виробником можна укласти контракт, щоб гарантувати, що це виконується без перешкод для регулярних операцій і що витрати на нього є низькими.

У проблемах поломки допоможуть постійні контракти з виробником, оскільки вони зможуть швидко реагувати та вирішувати проблеми в момент їх виникнення. Затримки у справі можуть бути не більше кількох хвилин, залежно від величини поломки.

У свою чергу основною перевагою автоматизованої системи керування паркуванням є ефективність, адже система не покладається на роботу людини; натомість мова йде про програмне забезпечення та смартфони, які менш схильні до помилок. Чоловік біля воріт має справу з великою кількістю людей одночасно; він може втомитися і навіть захворіти або відсутній на роботі. З іншого боку, автоматизоване програмне забезпечення завжди працює, поки у вас увімкнений телефон, що досить часто. Наступною

перевагою можна назвати надавання інформації за допомогою веб-додатку. Для сучасної системи, яка використовує програмне забезпечення, ви можете задавати різні питання через телефон. Маєте запитання щодо місць, зайнятості, тривалості перебування чи незаконної парковки? На них можна відповісти, де б ви не були. Автоматизована парковка має високу пропускну здатність та надійність при мінімальному складі обслуговуючого персоналу [20].

### 3.2 Розробка веб-дизайну

Ключовим аспектом у створенні автоматизованої системи автопаркінгу є веб-додаток. За допомогою додатку, водій може переглянути усю інформацію про систему послуг, наявність вільних місць та буде мати можливість забронювати одне.

Насамперед необхідно розробити головну сторінку, яка буде відповідати за візитівку автопаркінгу, та надавати необхідну інформацію клієнту про послуги (рис. 3.1).

Для розробки було використано: HTML, CSS та JS. За допомогою HTML було створено розмітку веб-додатку, що включає в себе головне меню, у вигляді нумерованого списку, заголовки, параграфи, що поділені на окремі секції. Сторінка складається з шапки, основного контенту та підвалу. Містить посилання на інші сторінки, що знаходять у кнопках з надписами. Головне меню є активним і при натисненні переведе користувача на необхідну секцію з детальною інформацією.

За допомогою CSS було розроблено стилі веб-додатку, а саме позиціонування елементів, що включає в себе такі засоби створення макетів, як grid, flexbox та float. Усі кнопки є активними та динамічними, при наведенні на них вони змінюють колір та курсор.

JavaScript виконує ключову роль у самому бронюванні вільних місць у додатку. За допомогою JS паркувальні місця можуть змінювати колір після натискання на них.

Головне меню додатку складається з п'яти частин. Розглянемо їх детальніше.

“Головна сторінка” — при натисненні завжди буде повертати на початкову сторінку сайту. “Про нас” — при натисненні переведе у блок з детальною інформацією про систему автопаркінгу. “Послуги” — при натисненні переведе у блок з умовами користування парковкою, що буде включати вартість користування, умови договору, та спосіб оплати. “Контакти” — блок, що містить контактний телефон директора, корпоративну пошту та адреси керівництва. “Парковки” — блок, що містить карту міста з усіма доступними парковками.

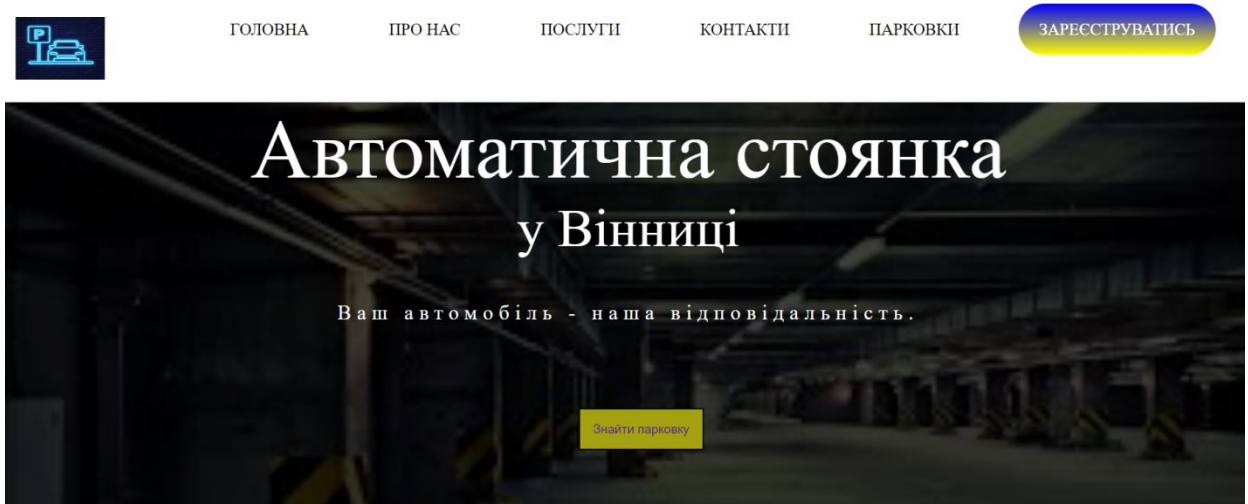


Рисунок 3.1 — Головне меню веб-додатку

Наступний розділ включає усі переваги парковки: відеонагляд, нові сучасні технології, графік роботи, усі доступні акції та можливість безкоштовного користування парковкою. Задача цієї секції запевнити користувача, що обравши наш додаток, та скориставшись саме нашою парковкою він отримає чудове обслуговування та в результаті чого буде дуже задоволений (рис. 3.2), адже він не тільки заощаджує свій дорогоцінний час, та ще й бензин, а як усім відомо, викиди в атмосферу чадного газу дуже шкодить нашій планеті і є основною причиною створення парникового ефекту.



## ПРО НАС

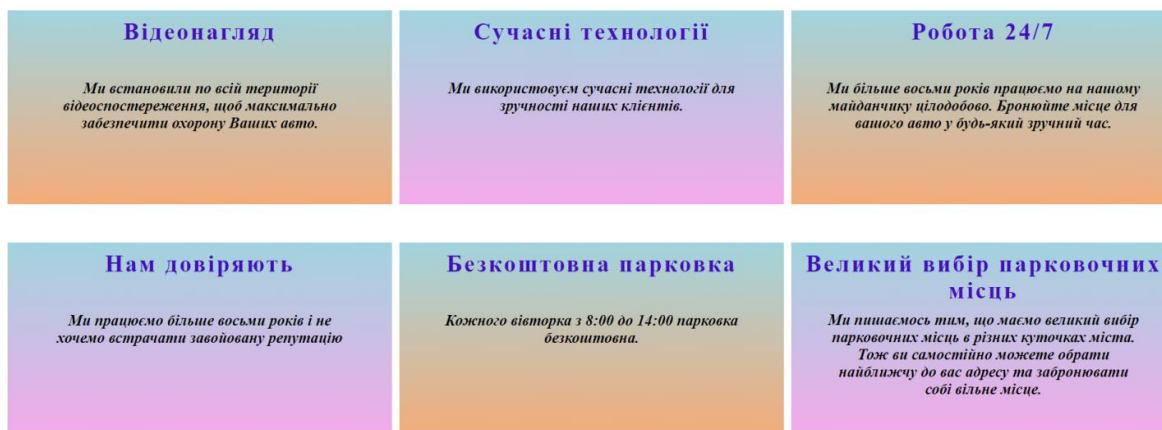


Рисунок 3.2 — Секція з перевагами парковки

На рисунку зображено 6 блоків, які включають в себе заголовок, що у свою чергу привертає увагу клієнта та сама інформація, яка виділена курсивом та жирним шрифтом. Задній фон блоків замальований за допомогою градієнта, що плавно переходить з одного кольору у інший.

Ознайомившись з додатком, водія може зацікавити як саме працює ця парковка та який принцип її роботи. Тож розділ під назвою “Послуги”, допоможе йому у цьому (рис. 3.3).

## ПОСЛУГИ

*Принципом роботи автоматизованих парковок є такий підхід: під час поїздки ви вирішили припаркуватись. Першим кроком є бронювання вільного місця на парковці. Вам потрібно зайти на наш веб-додаток, обрати найближчу до нього парковку та якщо є вільне місце, забронювати його на деякий час. За це час ви повинні доїхати до вибраної вами адреси. Після натискання кнопки “забронювати”, вам на пошту прийде лист з QR-кодом.*

*При в'їзді на парковку ви повинні зупинитися біля пристрою читання QR-кодів та показати його зчитувачу. Система перевіряє чи дійсний та коректний QR-код і відкриває шлагбаум. Якщо несподівано, з якоїсь причини, автомобіль зупиниться під шлагбаумом, шлагбаум не вдарить по машині, так як індукційні петлі відчують перебування автомобіля під стрілою шлагбауму, а також вдарити не дасть другий спутніть безпеки у вигляді ІК давача. У випадку, якщо QR-код не дійсний шлагбаум не підніметься.*

*Ви займаєте обране вами місце таке як і у додатку. Весь час поки автомобіль перебуває на стоянці, за ним спостерігає кілька відеокamer. При виїзді ви повинні знову показати свій QR-код системі, та система зчитуючи його вдруге розуміє, що ви хочете виїхати і відкриває шлагбаум, після цього з вашої карти автоматично стишують кошти. Картку вам необхідно прив'язати до додатку під час реєстрації.*

Рисунок 3.3 — Інформація про послуги, що надає інформаційна система обслуговування автопаркінгу

Ця секція включає детальне пояснення роботи роботи парковки, а саме як розраховуватись, як бронювати вільне місце та через який час його необхідно зайняти, тощо. Водію залишається лише уважно все прочитати та при реєстрації натиснути на кнопку згоди з умовами.

На головній сторінці міститься карта міста, на якій червоними прапорцями зображенні усі доступні парковки за різними адресами. Ця карта лише ознайомлює водія, та дає мінімальне представлення де знаходяться парковки (рис. 3.4). Для того щоб правильно користуватись картою, слід натиснути на неї, і згодом вона автоматично підтягнеться з Google Map.

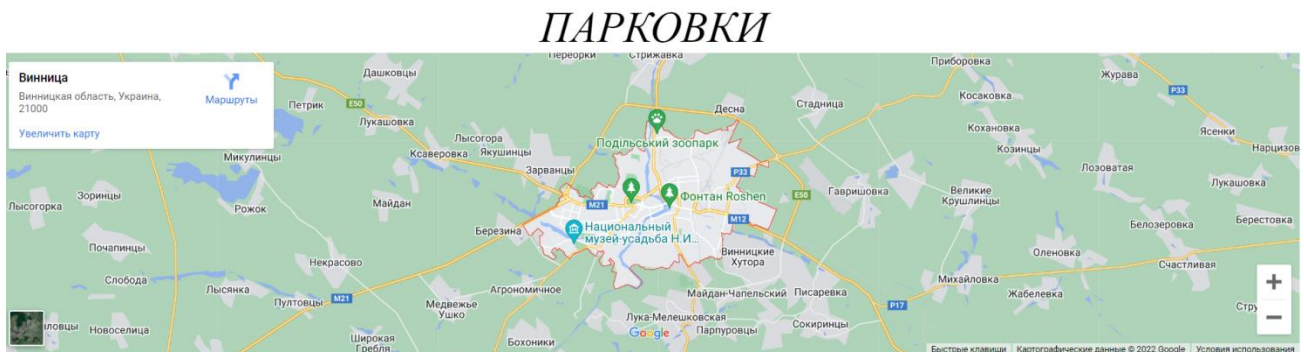


Рисунок 3.4 — Карта міста з позначеними на ній парковками

Якщо у клієнта виникнуть якісь питання, він завжди може звернутись за контактними номерами телефону, або написати на корпоративну пошту. Уся детальна інформація знаходиться внизу головної сторінки (рис 3.5).

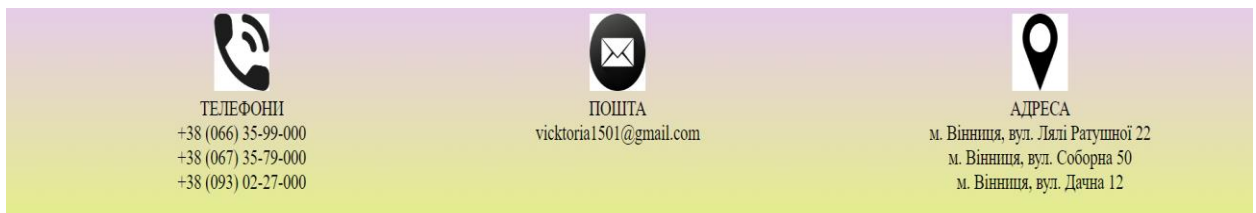


Рисунок 3.5 — Контактна інформація

### 3.3 Тестування веб-додатка

Тестування відіграє життєво важливу роль в процесі розробки і створення якісного програмного забезпечення. Необхідно серйозно ставитися

до аналізу і проєктування структурованого процесу, який забезпечить своєчасний і успішний випуск проєкту. Важливо пам'ятати, що довіру користувачів дуже просто втратити, а виправити допущені помилки може коштувати дорожче, ніж спочатку провести повну підготовку та тестування.

До тестування верстки входить візуальна частина, а саме:

- невірне відображення блоків, складових інтерфейсу, нестиковки колірної гами;
- тестування локалізованих версій (переклад сайту);
- відповідність макету (шари у PhotoShop);
- у разі зменшення/збільшення масштабів (75-150%) без візуальних недоліків;
- підсвічування полів з помилками;
- перевірка у дозволах (+ прокрутка).

До тестування JS входять перевірки:

- чи натискаються клікабельні елементи (внутрішні/зовнішні посилання, посилання на електронну пошту, кнопки, іконки);
- під час наведення на клікабельні елементи курсор змінюється, в іншому випадку – ні;
- підказки на незрозумілих клікабельних елементах;
- під час відключення зображень повинні бути підписи невеликим сірим кольором.

Під час функціональне тестування, необхідно перевірити:

- коректність роботи головних функцій сайту;
- перехід за посиланнями;
- перевірка користувальницьких форм (валідація полів, обов'язкові/необов'язкові поля, повідомлення про помилки під час неправильного введення, зворотний зв'язок);
- перевірка можливої авторизації/реєстрації.

Для того, щоб переконатись, що веб-додаток є зручним та вірно функціонує нам необхідно уявити себе водієм який вперше його відкрив.

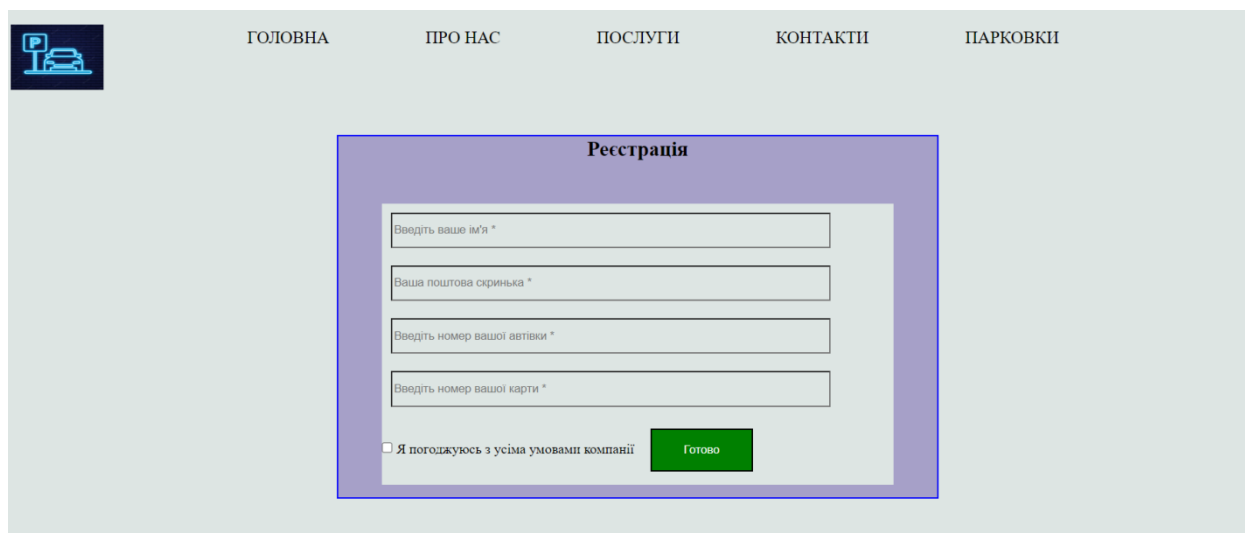
Отож, при завантаженні додатку відкривається головна сторінка, вміст якої було розглянуто у підрозділі 3.2.

При реєстрації необхідно вказати ім'я та прізвище користувача. Ця інформація необхідна для авторизації у майбутньому. Так як уся система побудована на принципі відправлення листа з QR-кодом на електронну скриньку, то пошта є обов'язковою складовою реєстрації.

Також водія попросять ввести номер своєї автівки. Ця інформація необхідна у справах безпеки, та у випадку викрадення автомобіля з парковки. І останнє, що необхідно це ввести номер своєї карти, аби прив'язати її до додатку. І коли водій покине парковку, гроші автоматично будуть списані з його карти.

Отож, ознайомившись з усією даною інформацією, заповнивши усі поля у формі та погоджуючись з усіма умовами, можна натискати кнопку "Готово". Першим кроком користування парковкою є реєстрація у додатку, якщо це раніше не було зроблено. Для цього слід натиснути на кнопку "Зареєструватись", що знаходиться у головному меню зверху сторінки.

Після натискання завантажується сторінка з формою для реєстрації (рис. 3.6).



The image shows a web page with a navigation menu at the top containing links for 'ГОЛОВНА', 'ПРО НАС', 'ПОСЛУГИ', 'КОНТАКТИ', and 'ПАРКОВКИ'. The main content area features a registration form titled 'Реєстрація'. The form includes four text input fields with the following labels: 'Введіть ваше ім'я \*', 'Ваша поштова скринька \*', 'Введіть номер вашої автівки \*', and 'Введіть номер вашої карти \*'. At the bottom of the form, there is a checkbox with the text 'Я погоджуюсь з усіма умовами компанії' and a green button labeled 'Готово'.

Рисунок 3.6 — Поле для реєстрації

Після того як водій ввів своє ім'я, адресу поштової скриньки, номер автівки та номер карти, з якої будуть списуватись кошти і погодився з усіма умовами компанії, самі умови написані на головній сторінці у секції послуги, йому слід натиснути на кнопку “Готово”. Тоді з'явиться повідомлення про те що вас було зареєстровано, та ви увійдете у свій обліковий запис. Вітаємо! Тепер ви можете користуватись усіма доступними послугами автопаркінгу.

Для того щоб, вийти зі свого облікового запису, необхідно натиснути на кнопку “Вихід”, що буде замінена замість кнопки “Зареєструватись” на головній сторінці.

Повертаємось на головну сторінку. Тепер нам необхідно знайти всі доступні парковки та дізнатись їхню адресу. Натискаємо на кнопку “Знайти парковку”. Завантажується сторінка з усіма доступними адресами парковок (рис. 13.7 ).

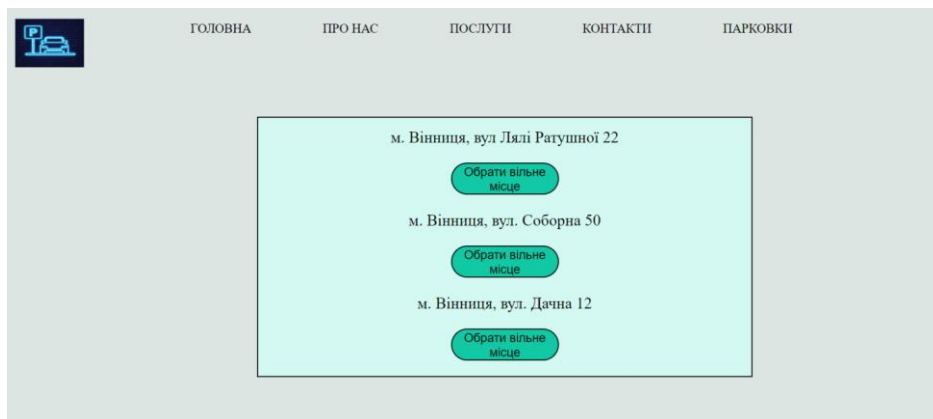


Рисунок 3.7 — Перелік доступних парковок

Після завантаження сторінки, яка зображена на рисунку 3.7, водій має обрати найближчу доступну йому адресу та натиснути на кнопку “Обрати вільне місце”. Адреси парковок будуть доповнюватись та розширяться. Якщо користувач хоче повернутись на головну сторінку, йому слід натиснути на кнопку “Головна”, або ж на любую іншу з усіх доступних. Після натиснення завантажувється сторінка з усіма паркувальними місцями (рис. 3.8).



Рисунок 3.8 — Сторінка з усіма паркувальними місцями

Червоним кольором позначені місця, які вже є зайнятими. Їх займати не можна та при натисненні нічого не відбудеться. Помаранчевим кольором позначені місця, які були заброньовані, іншими водіями, проте тут є особливість, якщо водій все ж не займе своє місце, то квадрат замалюється назад у зелений, і буде вважатись як вільний, проте якщо водій за зазначений час встигне зайняти своє місце, то квадрат замалюється у червоний, що буде означати — зайнято.

Отож, водію можна натискати лише на зелені квадрати. При натисненні на квадрат, з'явиться повідомлення з підтвердженням бронювання місця (рис. 3.9). Якщо водій натисне — “Так”, місце зафарбується у жовтий колір та йому на пошту прийде лист з QR-кодом, який він має показати зчитувачу при в'їзді, якщо ж водій обирає інший варіант, його повертає на головну сторінку (рис. 3.10).

Якщо ж водій забронював собі місце, але не зайняв його впродовж 25 хвилин, жовтий квадрат змінить свій колір на зелений, що буде свідчити про ще одне звільнене місце. Коли водій буде виїжджати з парковки, і при цьому при виїзді покаже ще раз QR-код, червоний квадрат автоматично змінить свій колір на зелений.



Рисунок 3.9 — Приклад повідомлення з підтвердженням бронювання

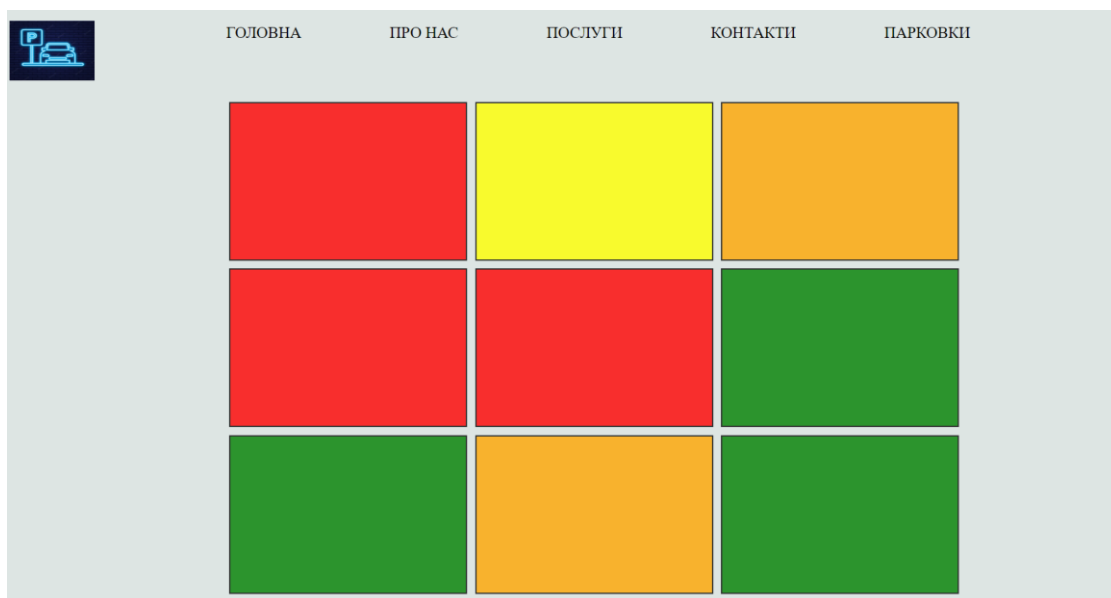


Рисунок 3.10 — Приклад відображення сторінки, після бронювання одного з місць

Отож, у даному додатку вірне відображення блоків, складових інтерфейсу, колірна гама підібрана вдало, кольори використані теплі та комфортні для користувацького ока. Сайт містить лише українську мову, проте у розробці переклад ще на англійську мову. У разі зменшення та збільшення масштабів візуальних недоліків немає. Додаток є адаптивним як для планшету так і для телефону. Усі клікабельні елементи, а саме : кнопки,

пошта, телефон, іконки, працюють коректно та містять усі необхідні посилання. Під час наведення на клікабельні елементи курсор змінюється. Підказки на незрозумілих клікабельних елементах присутні. Отже, даний додаток є доволі зручним у використанні, містить приємні теплі кольори, градієнти, рисунки, що привертають увагу користувача та тим самим зацікавлюють. Сам додаток є інтуїтивно простим, з ним можуть справитись усі водії не залежно від їхнього віку.



## ВИСНОВОК

У ході виконання бакалаврської роботи були досліджені та випробувані методи створення веб-додатку інформаційного обслуговування автопаркінгу.

У першому розділі було проаналізовано сучасні підходи до побудови інформаційних систем паркінгу, наведено ряд переваг та недоліків таких парковок. Розглянуто різницю між автоматичними та напівавтоматичними системами. Проаналізовано усі можливі аналоги таких систем.

У другому розділі було розглянуто поняття різниці між веб-сайтом та веб-додатком, було наведено порівняльну характеристику у вигляді таблиці. Проаналізовані переваги та недоліки веб-додатків та описано принципи їх роботи. Для того щоб створити веб-додаток, необхідно використовувати HTML, CSS та JS, тож було розглянуто їх теоретичні основи.

У третьому розділі було розв'язано всі поставлені задачі: створено веб-додаток для інформаційного обслуговування автопаркінгу, який включає можливість переглядання та бронювання вільних паркувальних місць. Реалізовано зручний, інтуїтивний інтерфейс та функціонал, що дозволяє легко користуватись додатком.

Отже, у результаті дипломної роботи було отримано працездатний веб-додаток, який можна використовувати як систему інформаційного обслуговування автопаркінгу. Беззаперечно цей проект буде і надалі розвиватись, в планах розвивати готовий функціонал, додавати новий, робити можливість інтегрування проекту у інші, покращувати зовнішній вигляд та зручність у використанні.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ягузинська І. Ю., Типушова І. О. Сучасні автоматизовані системи паркування автомобілів // Науково-методичний електронний журнал “Концепт” – 2020. – Т. 35. – С. 156–160. – URL: <http://e-koncept.ua/2015/95585.htm>.
2. Інновації у парковках: які технології можуть допомогти міській інфраструктурі? [Електронний ресурс]. Режим доступу: <https://www.forbes.ua/tehnologii/340561-innovacii-v-parkovkah-kakie-tehnologii-mogut-pomoch-gorodskoy-infrastrukture>.
3. Сучасні паркувальні системи. [Електронний ресурс]. Режим доступу: <https://dentauto.ua/raznoe/sovremennye-parkovochnye-sistemy.html>.
4. Розумна парковка та її переваги [Електронний ресурс]. Режим доступу: <https://ula.lantec.ua/statti/rozumna-parkovka-ta-jiji-perevagi>
5. Игнатъев Ю.В. Возведение автомобильных стоянок и парковок в крупных городах //К: 2019. № 17. С. 68–72.
6. Автоматизовані системи платного паркування [Електронний ресурс]. <https://parkmag./news/obzor-avtomatizirovannyh-sistem-platnoy-parkovki.html>
7. Системи управління парковками [Електронний ресурс] – режим доступу: <https://ia.ua/solutions-ru/infrastructure-object/building-automation/199-parking-control-systems/>
8. Автоматизована система парковки [Електронний ресурс] – режим доступу: <https://lotgroup.ua/product/smart-city/automated-parking-system/>
9. Автоматизована система парковки [Електронний ресурс] – режим доступу: [https://idcard.com.ua/solutions/parking\\_rfid/](https://idcard.com.ua/solutions/parking_rfid/)
10. Автоматизована система парковки [Електронний ресурс] – режим доступу: <http://rfidukraine.com.ua/automation-of-parking-based-on-uhf-technology/>

11. Web application [Електронний ресурс] – режим доступу: <https://www.techtarget.com/searchsoftwarequality/definition/Web-application-Web-app>
12. Websires vs web application [Електронний ресурс] – режим доступу: <https://www.wix.com/velo/blog/post/websites-vs-web-applications>
13. Розробка веб-застосунків [Електронний ресурс] — режим доступу: <https://webstudio2u.net/ua/site-develop/641-razrabotka-veb-prilozheniy.html>.
14. Кучай А.В., Білан С.О. «Основи HTML» //К: 2018. № 12.С. 20-25.
15. What is HTML? [Електронний ресурс] – режим доступу: <https://www.hostinger.com/tutorials/what-is-html>
16. Основи CSS [Електронний ресурс] – режим доступу: <https://www.w3.org/Style/Examples/007/units.en.html>
17. CSS для початківців [Електронний ресурс] – режим доступу: [https://www.tutorialspoint.com/css/css\\_measurement\\_units.htm](https://www.tutorialspoint.com/css/css_measurement_units.htm)
18. What is JavaScript [Електронний ресурс]. Режим доступу: <https://www.tutorialsteacher.com/javascript/what-is-javascript> Дата звернення: Травень 21, 2020.
19. JS Theories [Електронний ресурс] – режим доступу: [https://is.theorizeit.org/wiki/Main\\_Page](https://is.theorizeit.org/wiki/Main_Page)
20. Pros and cons of parking management systems [Електронний ресурс] – режим доступу: <https://www.bookingninja.com/blog/pros-and-cons-of-parking-management-systems>

## ДОДАТОК А

Технічне завдання

Міністерство освіти і науки України  
Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ  
проф., д.т.н.. Азаров О.Д..  
" " 2022 р.

### ТЕХНІЧНЕ ЗАВДАННЯ

на виконання бакалаврської дипломної роботи  
“Веб-додаток інформаційного обслуговування автопаркінгу”  
08-23.БДР.025.00.000 ТЗ

Науковий керівник: доцент к.т.н.

\_\_\_\_\_ Тарновський М. Г.

Студент групи 2КІ-186

\_\_\_\_\_ Кривенька В. О.

м. Вінниця – 2022

## 1 Підстава для використання бакалаврської кваліфікаційної роботи (БДР)

1.1 Актуальність розробки полягає у необхідності вирішення проблеми пошуку вільних місць на парковках, шляхом розробки веб-додатку, головною ціллю якого є використання автоматизованої паркувальної системи, що у свою чергу спрощує пошук та дозволяє бронювати вільні місця.

1.2 Наказ про затвердження теми бакалаврської дипломної роботи.

## 2 Мета і призначення БДР

2.1 Мета проекту — розробка зручного у використанні веб-додатку інформаційного обслуговування автопаркінгу призначеного для загального користування.

2.2 Призначення розробки — виконання бакалаврського дипломного проекту із подальшим впровадженням та розвитком продукту.

## 3 Вихідні дані для виконання БДР

3.1 Проведення аналізу сучасних підходів до побудови інформаційних систем паркінгу.

3.2 Розгляд принципів роботи автоматизованих паркувальних систем.

3.2 Розробка інтерфейсу та функціоналу веб-додатку.

## 4 Вимоги до виконання БДР

Головна вимога — розробити зручний інтерфейс до веб-додатку, використовуючи усі здобуті навички та знання.

## 5 Етапи БДР та очікувані результати

Етапи роботи та очікувані результати приведено в Таблиці А.1.

## 6 Матеріали, що подаються до захисту БДР

До захисту подаються: пояснювальна записка БДР, ілюстративні

матеріали, протокол попереднього захисту БДР на кафедрі, відгук наукового керівника, анотації до БДР українською та іноземною мовами, довідка про відповідність оформлення БДР діючим вимогам.

Таблиця А.1 — Етапи БДР

№ з/п	Назва етапів виконання комплексної бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Постановка задачі роботи	07.03.22	Аналітичний огляд літературних джерел, розділ 1
2	Аналіз предметної області	08.03-18.03.22	розділ 1
3	Аналіз сучасних підходів до побудови інформаційних систем паркінгу	21.03-31.03.22	розділ 1
4	Аналіз поняття веб-додатку	04.04-15.04.22	розділ 2
5	Аналіз принципів роботи веб-додатків	18.04-22.04.22	розділ 2
6	Проектування веб-додатку	25.04-.06.05.22	розділ 3
7	Розробка веб-дизайну	09.05-13.05.22	розділ 3
8	Розробка функціоналу веб-додатку	16.05-20.05.22	розділ 3
9	Підготовка матеріалів та опис розробки інформаційної системи	23.05-26.05.22	Тези доповідей
10	Аналіз виконання роботи, висновки, додатки	27.05-31.05.22	Пояснювальна записка
11	Перевірка якості виконання бакалаврського проекту та усунення недоліків	01.06 -.08.06.22	Пояснювальна записка, графічний матеріал і презентація

## 7 Порядок контролю виконання та захисту БДР

Виконання етапів графічної та розрахункової документації БДР контролюється науковим керівником згідно зі встановленими термінами. Захист БДР відбувається на засіданні Екзаменаційної комісії, затвердженої наказом ректора.

## 8 Вимоги до оформлювання та порядок виконання БДР

При оформлюванні БДР використовуються:

- ДСТУ 3008 : 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;
- ДСТУ 8302 : 2015 «Бібліографічні посилання. Загальні положення та правила складання»;
- ГОСТ 2.104-2006 «Єдина система конструкторської документації. Основні написи»;
- документами на які посилаються у вище вказаних.

## ДОДАТОК Б

### Лістинг HTML файлу

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Веб-застосунок</title>
<link rel="stylesheet" href="css/style.css">
</head>
<body>
<header>
<nav>

<ul>
<li><a class="href2" href="">Головна</a> </li>
<li><a class="href2" href="">Про нас</a></li>
<li><a class="href2" href="">Послуги</a> </li>
<li><a class="href2" href="">Контакти</a> </li>
<li><a class="href2" href="">Парковки</a></li>
<li><a class="href2" href="registration.html">Зареєструватись</a></li>
</ul>
</nav>
</header>
<main>
<div class="header">
<p class="header_text"> Автоматична стоянка </p>
<p class="header_text2">у Вінниці</p>
```



```

<p class="header_text3">Ваш автомобіль - наша відповідальність. </p>
<button class="button_parkovka"><a class="href3" href="parkovka.html">Знайти
парковку</a></button>
</div>
<h2 class="about_us">Про нас </h2>
<div class="about">
<div class="park">
<h3 class="park_h">Відеонагляд</h3>
<div class="park_text">Ми встановили по всій території відеоспостереження,
щоб максимально забезпечити охорону Ваших авто.</div>
</div>
<div class="park_2">
<h3 class="park_h">Сучасні технології</h3>
<div class="park_text"> Ми використовуємо сучасні технології для зручності
наших клієнтів. </div>
</div>
<div class="park">
<h3 class="park_h">Робота 24/7</h3>
<div class="park_text"> Ми більше восьми років працюємо на нашому
майданчику цілодобово. Бронуйте місце для вашого авто у будь-який
зручний час.</div>
</div>
</div>
<div class="about">
<div class="park_2">
<h3 class="park_h">Нам довіряють</h3>
<div class="park_text"> Ми працюємо більше восьми років і не хочемо
втрачати завойовану репутацію</div>
</div>

```

```
<div class="park">
```

```
<h3 class="park_h">Безкоштовна парковка</h3>
```

```
<div class="park_text"> Кожного вівторка з 8:00 до 14:00 парковка  
безкоштовна.</div>
```

```
</div>
```

```
<div class="park_2">
```

```
<h3 class="park_h">Великий вибір парковочних місць</h3>
```

```
<div class="park_text_2">Ми пишаємось тим, що маємо великий вибір  
парковочних місць в різних куточках міста. Тож ви самостійно можете  
обрати найближчу до вас адресу та забронювати собі вільне місце.</div>
```

```
</div>
```

```
</div>
```

```
<div class="poslugy">
```

```
<h3 class="about_us"> Послуги </h3>
```

```
<p class="poslugy_text"> Принципом роботи автоматизованих парковок є  
такий підхід: під час поїздки ви вирішили припаркуватись. Першим кроком є  
бронювання вільного місця на парковці. Вам потрібно зайти на наш веб-  
додаток, обрати найближчу до нього парковку та якщо є вільне місце,  
забронювати його на деякий час. За це час ви повинні доїхати до вибраної  
вами адреси. Після натискання кнопки “забронювати”, вам на пошту прийде  
лист з QR-кодом. <p class="poslugy_text">При в'їзді на парковку ви повинні  
зупинитися біля пристрою читання QR-кодів та показати його зчитувачу.  
Система перевіряє чи дійсний та коректний QR-код і відкриває шлагбаум.  
Якщо несподівано, з якоїсь причини, автомобіль зупиниться під шлагбаумом,  
шлагбаум не вдарить по машині, так як індукційні петлі відчують  
перебування автомобіля під стрілою шлагбауму, а також вдарити не дасть  
другий ступінь безпеки у вигляді ІК давача. У випадку, якщо QR-код не  
дійсний шлагбаум не підніметься.</p>
```

```
<p class="poslugy_text">Ви займаєте обране вами місце таке як і у додатку.  
Весь час поки автомобіль перебуває на стоянці, за ним спостерігає кілька
```

відеокамер. При виїзді ви повинні знову показати свій QR-код системі, та система зчитуючи його вдруге розуміє, що ви хочете виїхати і відкриває шлагбаум, після цього з вашої карти автоматично спишуться кошти. Картку вам необхідно прив'язати до додатку під час реєстрації. </p>

</p>







</div>

<div class="parkovka">

<p class="about\_us"> Парковки </p>

<iframe

src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d83366.18447108139!2d28.39959399457278!3d49.23482492015501!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x472d5b65195a6489%3A0xcbd4e159eedd74fe!2z0JLQuNC90L3QuNGG0LAsINCS0LjQvdC90LjRhtC60LDRjyDQvtCx0LvQsNGB0YLRjCwg0KPQutGA0LDQuNC90LAsIDIxMDAw!5e0!3m2!1sru!2snl!4v1652204477483!5m2!1sru!2snl" width="1510" height="350" style="border:0;" allowfullscreen="" loading="lazy" referrerpolicy="no-referrer-when-downgrade"></iframe>

</div>

</main>

<footer class="contact">

<div class="display">

<div class="foo">



<p> ТЕЛЕФОНИ <br>

+38 (066) 35-99-000

<br>

+38 (067) 35-79-000

```
<br>
+38 (093) 02-27-000</p>
</div>
<div class="foo">

<p> ПОШТА<br>
vicktoria1501@gmail.com</p>
</div>
<div class="foo">

<p>АДРЕСА <br>
м. Вінниця, вул. Лялі Ратушної 22 <br>
м. Вінниця, вул. Соборна 50 <br>
м. Вінниця, вул. Дачна 12 <br>
</p>
</div>
</div>
</footer>
</body>
</html>
```

## ДОДАТОК В

### Лістинг CSS файлу

```
*{
  margin:0;
}
.href2{
  text-decoration: none;
  color:inherit;
  padding: 20px 25px;
}
.href2:hover{
  background: linear-gradient(blue, yellow);
  transition: background-color 0.9s;
  border-radius: 45px;
  color:white;
}
.href3{
  text-decoration: none;
}
ul{
  display: flex;
  text-indent: 3vw;
  font-family: 'Montserrat';
  font-weight: 400;
  font-size: 1.3vw;
  text-transform: uppercase;
  position: absolute;
  left: 170px;
  top: 27px;
```

```
}  
li {  
  list-style-type: none;  
  display: inline-block;  
}  
.logo{  
margin: 1.5vw;  
width: 7vw;  
height:10vh;  
}  
.header{  
  background-image: url("../images.jpg");  
  background-repeat: no-repeat;  
  background-size: cover;  
  height:500px;  
  box-shadow:0 0 0 400px rgba(0, 0, 0, 0.75) inset;  
}  
.header_text{  
  color:white;  
  text-align: center;  
  font-size: 100px;  
}  
.header_text2{  
  color:white;  
  text-align: center;  
  font-size: 65px;  
}  
.header_text3{  
  margin-top: 50px;  
  color:white;
```

```
    text-align: center;
    font-size: 25px;
    letter-spacing: 0.27em;
}
.button_parkovka{
    margin-left: 700px;
    margin-top: 100px;
    width: 150px;
    height:50px;
    font-size: 15px;
    background-color: #A4A211;
}
.button_parkovka:hover{
    background-color: #A45411;
}
.about{
    display: flex;
}
.about_us{
    text-align: center;
    margin-top:25px;
    text-transform: uppercase;
    font-size: 50px;
    color:black;
    font-style: italic;
}
.park{
    margin:20px 5px 30px;
    width:600px;
    height: 250px;
```

```
    background: linear-gradient(#A2D4DE,#F4AC7A);
}
.park_2{
    margin:20px 5px 30px;
    width:600px;
    height: 250px;
    background: linear-gradient(#A2D4DE,#F4AAEB);
}
.park:hover{
    margin-top: 10px;
    transition: all 0.5s ease;
    background: linear-gradient(#A2D4DE,#E2F99A);
}
.park_2:hover{
    margin-top: 10px;
    transition: all 0.5s ease;
    background: linear-gradient(#A2D4DE,#E2F99A);
}
.park_h{
    color:#460CC1;
    text-align: center;
    font-size: 30px;
    margin: 10px;
    letter-spacing: 0.10em;
}
.park_text{
    text-align: center;
    padding: 10px;
    font-size: 20px;
    font-style: italic;
```



```
margin: 35px;
font-weight: bold;
}
.park_text_2{
text-align: center;
padding: 10px;
font-size: 20px;
font-style: italic;
margin: -10px 35px;
font-weight: bold;
}
.posluga{
margin: 50px 10px 25px;
height: 500px;
background: linear-gradient(#EFD9BD,#E4F085);
}
.posluga_text{
text-align: center;
font-size: 20px;
margin: 15px 25px;
font-style: italic;
}
.posluga_img{
margin-left: 300px;
}
.parkovka{
height: 350px;
}
.contact{
margin-top:85px;
```

```
background: linear-gradient(#E5CAEE,#E4F085);
height: 200px;
}
.display{
display: flex;
}
.icons{
width: 65px;
height: 65px;
}
.foo{
margin:15px 50px 2px;
margin-left: 220px;
text-align: center;
font-size: 18px;
}
```

## ДОДАТОК Г

### Лістинг JS файлу

```
function myFunc2() {
  confirm("Ви впевнені, що хочете забронювати це місце?");
  changeColor(2, 'yellow');
}
function myFunc6() {
  confirm("Ви впевнені, що хочете забронювати це місце?");
  changeColor(6, 'yellow');
}
function myFunc7() {
  confirm("Ви впевнені, що хочете забронювати це місце?");
  changeColor(7, 'yellow');
}
function myFunc9() {
  confirm("Ви впевнені, що хочете забронювати це місце?");
  changeColor(9, 'yellow');
}
function changeColor(number, color) {
  let hs = document.querySelectorAll('#col_' + number);
  for (let i = hs.length; i--;) {
    hs[i].style.background = color;
  }
}
```

## ДОДАТОК Д

### Головна сторінка веб-додатку


ГОЛОВНА
ПРО НАС
ПОСЛУГИ
КОНТАКТИ
ПАРКОВКИ
ЗАРЕЄСТРУВАТИСЬ

# Автоматична стоянка

## у Вінниці

Ваш автомобіль - наша відповідальність.

[Знайти парковку](#)

### ПРО НАС

#### Відеонагляд

Ми встановили по всій території відеоспостереження, щоб максимально забезпечити охорону Ваших авто.

#### Сучасні технології

Ми використовуємо сучасні технології для зручності наших клієнтів.

#### Робота 24/7

Ми більше восьми років працюємо на нашому майданчику плодотворно. Бронюйте місце для вашого авто у будь-який зручний час.

#### Нам довіряють

Ми працюємо більше восьми років і не хочемо втрачати завоювану репутацію

#### Безкоштовна парковка

Кожного вівторка з 8:00 до 14:00 парковка безкоштовна.

#### Великий вибір парковочних місць

Ми пинаємося тим, що маємо великий вибір парковочних місць в різних кутках міста. Тож ви самостійно можете обрати найближчу до вас адресу та забронювати собі вільне місце.

### ПОСЛУГИ

Принципом роботи автоматизованих парковок є такий підхід: під час поїздки ви вирішили припаркуватися. Перим кроком є бронювання вільного місця на парковці. Вам потрібно зайти на наш веб-додаток, обрати найближчу до нього парковку та якщо є вільне місце, забронювати його на деякий час. За це час ви повинні доїхати до вибраної вами адреси. Після написання кодуки "забронювати", вам на пошту прийде лист з QR-кодом.

При в'їзді на парковку ви повинні зупинитися біля пристрою читання QR-кодів та показати його зчитувачу. Система перевіряє чи дійсний та коректний QR-код і відкриває шлагбаум. Якщо несподівано, з якоїсь причини, автомобіль зупиниться під шлагбаумом, шлагбаум не відкриє по машині, так як індукційні петлі відчувать перебування автомобіля під стрілою шлагбауму, а також відкрили не дасть другий ступінь безпеки у вигляді ІК датчика. У випадку, якщо QR-код не дійсний шлагбаум не підніметься.

Ви зможете обрати вільне місце таке як і у додатку. Весь час поки автомобіль перебуває на стоянці, за ним спостерігає кілька відеонагляд. При в'їзді ви повинні знову показати свій QR-код системі, та система зчитуючи його відкриє розлізе, що ви хочете вийти і відкрити шлагбаум, після чого з вашої карти автоматично спинуться кошти. Картку вам необхідно прив'язати до додатку під час реєстрації.





### ПАРКОВКИ





**ТЕЛЕФОНІ**  
+38 (066) 35-99-000  
+38 (067) 35-79-000



**ПОШТА**  
vicktonia1501@gmail.com



**АДРЕСА**  
м. Вінниця, вул. Лева Ратушної 22  
м. Вінниця, вкл. Соборна 50

Рисунок Д.1 — Головна сторінка веб-додатку

**ДОДАТОК Е**  
**ПРОТОКОЛ**  
**ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ**  
**РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ**  
**ЗАПОЗИЧЕНЬ**

Назва роботи: Веб-додаток інформаційного обслуговування автопаркінгу

Тип роботи: бакалаврська дипломна робота  
 (БДР, МКР)

Підрозділ кафедра обчислювальної техніки  
 (кафедра, факультет)

**Показники звіту подібності**  
**Unicheck**

Оригінальність 80,4% Схожість 19,6%

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку \_\_\_\_\_  
 (підпис)

Захарченко С.М.  
 (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи \_\_\_\_\_

Кривенька В. О.

Керівник роботи \_\_\_\_\_

Тарновський М. Г.