

ДОДАТОК А

Лістинг програми

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Kharkiv Smart Parking</title>
  <link rel="stylesheet" href="css/style.css">
</head>
<body>
  <div id="admin-indicator" style="display: none; background-color: darkgreen;
padding: 5px; border-radius: 5px;">
    <span style="color: rgb(251, 255, 0); font-size: 24px;">АДМІН</span>
  </div>

  <header>
    <h1>Kharkiv Smart Parking</h1>
  </header>
  <nav id="nav-bar" style="display:none;">
    <ul>
      <li><a href="#" onclick="showPage('home')">Головна сторінка</a></li>
      <li><a href="#" onclick="showPage('about')">Про нас</a></li>
      <li><a href="#" onclick="showPage('parking20')">Парковка для
МОТОЦИКЛІВ</a></li>
      <li><a href="#" onclick="showPage('parking40')">Парковка для
автомобілів</a></li>
      <li><a href="#" onclick="showPage('parkingBike')">Парковка для
велосипедів</a></li>
    </ul>
```

```

</nav>
<main>
  <section id="login" class="content-section">
    <h2>Авторизація</h2>
    <form id="loginForm">
      <label for="username">Логін:</label>
      <input type="text" id="username" name="username" required>
      <label for="password">Пароль:</label>
      <input type="password" id="password" name="password" required>
      <button type="submit">Увійти</button>
    </form>
    <button onclick="showRegistration()">Реєстрація</button>
    <button onclick="guestLogin()">Увійти як гість</button>
  </section>
  <section id="registration" class="content-section" style="display:none;">
    <h2>Реєстрація</h2>
    <form id="registrationForm">
      <label for="newUsername">Логін:</label>
      <input type="text" id="newUsername" name="newUsername" required>
      <label for="newPassword">Пароль:</label>
      <input type="password" id="newPassword" name="newPassword"
required>
      <button type="submit">Зареєструватися</button>
    </form>
    <button onclick="showLogin()">Вже зареєстровані? Увійти</button>
  </section>
  <section id="home" class="content-section" style="display:none;">
    <h2>Головна сторінка</h2>
    <p class="large-text">Цей повністю безкоштовний сайт допоможе Вам
знайти вільне паркувальне місце на парковках,

```

які підключені до нашої системи розумного паркування. Парковки, під'єднані до нашої системи, відображені на карті знизу.</p>

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

```
</section>
```

```
<section id="about" class="content-section" style="display:none;">
```

```
<h2>Про нас</h2>
```

```
<p class="large-text">Ми використовуємо технологію ультразвукових датчиків для перевірки зайнятості стоянкових місць,
```

```
щоб зекономити ваш час під час пошуку місць для паркування та надати інформацію з мінімальною затримкою, щоб уникнути ситуацій,
```

```
коли Ви приїжджаєте на зайняте місце Ви можете його забронювати для себе. Ми додаємо до системи всі бажані компанії,
```

```
які хочуть приєднати свої парковки до загальної мережі.</p>
```

```

```

```
</section>
```

```
<section id="parking20" class="content-section" style="display:none;">
```

```
<h2>Парковка для мотоциклів</h2>
```

```
<div class="parking-grid" id="grid20"></div>
```

```
<form id="bookingForm20">
```

```
<label for="parkingSpace20">Номер місця:</label>
```

```
<input type="number" id="parkingSpace20" name="parkingSpace20" min="1" max="20" required>
```

```
<label for="bookingTime20">Час бронювання (хв):</label>
```

```
<input type="number" id="bookingTime20" name="bookingTime20" min="1" required>
```

```
<button
```

```
type="button"
```

```
onclick="bookParkingSpace(20)">Забронювати</button>
```

```
</form>
```

```

```

```

</section>
<section id="parking40" class="content-section" style="display:none;">
  <h2>Парковка для автомобілів</h2>
  <div class="parking-grid" id="grid40"></div>
  <form id="bookingForm40">
    <label for="parkingSpace40">Номер місця:</label>
    <input type="number" id="parkingSpace40" name="parkingSpace40"
min="1" max="40" required>
    <label for="bookingTime40">Час бронювання (хв):</label>
    <input type="number" id="bookingTime40" name="bookingTime40"
min="1" required>
    <button type="button"
onclick="bookParkingSpace(40)">Забронювати</button>
  </form>
  
</section>
<section id="parkingBike" class="content-section" style="display:none;">
  <h2>Парковка для велосипедів (40 місць)</h2>
  <div class="parking-grid" id="gridBike"></div>
  <form id="bookingFormBike">
    <label for="parkingSpaceBike">Номер місця:</label>
    <input type="number" id="parkingSpaceBike"
name="parkingSpaceBike" min="1" max="40" required>
    <label for="bookingTimeBike">Час бронювання (хв):</label>
    <input type="number" id="bookingTimeBike"
name="bookingTimeBike" min="1" required>
    <button type="button"
onclick="bookParkingSpace('Bike')">Забронювати</button>
  </form>

```

```

    
    </section>
</main>
<footer>
    <p>&copy; 2024 Kharkiv Smart Parking</p>
</footer>
<script src="js/script.js"></script>
<script src="js/sensorData.js"></script>
<script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyAIMN3C8CAPGS0JG
T8dZAWfKIfAqs-h2vY&callback=initMap" async defer></script>
</body>
</html>
<div id="login-info" style="display: none;"></div>

```

```
// script.js
```

```
// Функція для відображення вибраної сторінки
```

```
function showPage(pageId) {
```

```
    // Ховаємо всі секції контенту
```

```
    document.querySelectorAll('.content-section').forEach(function(section) {
        section.style.display = 'none';
    });
```

```
// Видаляємо клас "active" у всіх посилань в навігації
```

```
document.querySelectorAll('nav ul li a').forEach(function(link) {
    link.classList.remove('active');
});
```

```

// Відображаємо вибрану секцію
document.getElementById(pageId).style.display = 'block';

// Додаємо клас "active" для вибраної вкладки
if (pageId !== 'login' && pageId !== 'registration') {
    document.querySelector(`nav ul li
a[onclick="showPage('${pageId}')"]`).classList.add('active');
}
}

// Функція для створення сітки парковки
function createParkingGrid(gridId, places, sensorData, updateSensorData) {
    const grid = document.getElementById(gridId);
    grid.innerHTML = ""; // Очищення попередніх даних
    for (let i = 1; i <= places; i++) {
        const space = document.createElement('div');
        space.className = `parking-space ${sensorData[i - 1] ? 'free' : 'occupied'}`;
        space.dataset.space = i; // Додаємо атрибут data-space з номером місця
        space.innerText = i;
        grid.appendChild(space);
    }
}

// Ініціалізація
document.addEventListener('DOMContentLoaded', function() {
    showPage('login'); // Показати сторінку входу за замовчуванням

    // Обробник форми входу
    document.getElementById('loginForm').addEventListener('submit',
function(event) {

```

```

event.preventDefault();
const username = document.getElementById('username').value;
const password = document.getElementById('password').value;
const role = checkLogin(username, password);
if (role) {
    showNavBar();
    showPage('home');
    const expirationTime = new Date().getTime() + 10 * 60 * 1000; //

```

Поточний час + 10 хвилин

```

    localStorage.setItem('currentUser', username);
    localStorage.setItem('currentRole', role);
    localStorage.setItem('expirationTime', expirationTime);
    showLoggedInUser(username);
} else {
    alert('Неправильний логін або пароль.');
```

}

```

});

// Обробник форми реєстрації
document.getElementById('registrationForm').addEventListener('submit',
function(event) {
    event.preventDefault();
    const username = document.getElementById('newUsername').value;
    const password = document.getElementById('newPassword').value;
    saveUser(username, password);
    alert('Реєстрація успішна. Тепер ви можете увійти.');
```

showLogin();

```

});

```

```

createParkingGrid('grid20', 20, sensorDataMotorcycle,

```

```
updateMotorcycleSensorData);
    createParkingGrid('grid40', 40, sensorDataCar, updateCarSensorData);
    createParkingGrid('gridBike', 40, sensorDataBike, updateBikeSensorData);
    initMap();
});
```

```
function showNavBar() {
    document.getElementById('nav-bar').style.display = 'block';
}
```

// Показати сторінку реєстрації

```
function showRegistration() {
    showPage('registration');
}
```

// Показати сторінку входу

```
function showLogin() {
    showPage('login');
}
```

// Гостьовий вхід

```
function guestLogin() {
    showNavBar();
    showPage('home');
}
```

// Перевірка логіна та пароля

```
function checkLogin(username, password) {
    if (username === 'admin' && password === '88888888') {
        return 'admin';
    }
}
```

```

    }
    const users = JSON.parse(localStorage.getItem('users') || '[]');
    const user = users.find(u => u.username === username && u.password ===
password);
    return user ? 'user' : null;
  }

// Збереження нового користувача
function saveUser(username, password) {
  const users = JSON.parse(localStorage.getItem('users') || '[]');
  users.push({ username, password });
  localStorage.setItem('users', JSON.stringify(users));
}

// Ініціалізація карти
function initMap() {
  new google.maps.Map(document.getElementById("map"), {
    zoom: 13,
    center: { lat: 49.9935, lng: 36.2304 } // Координати Харкова
  });
}

// Функція для бронювання місця парковки
function bookParkingSpace(gridId) {
  const          spaceNumber          =
document.getElementById(` parkingSpace${ gridId }`).value;
  const          bookingTime          =
document.getElementById(` bookingTime${ gridId }`).value;
  const grid = document.getElementById(` grid${ gridId }`);
  const parkingSpace = grid.querySelector(`.parking-space[data-

```

```

space="${spaceNumber}"]`);
    if (parkingSpace.classList.contains('free')) {
        parkingSpace.classList.remove('free');
        parkingSpace.classList.add('occupied');
        console.log(`Місце ${spaceNumber} заброньовано на ${bookingTime}
хвилин`);
    } else {
        console.log(`Місце ${spaceNumber} вже зайнято`);
    }
}

// Перевірка наявності інформації про користувача при завантаженні сторінки
document.addEventListener('DOMContentLoaded', function() {
    const expirationTime = localStorage.getItem('expirationTime');
    if (expirationTime && new Date().getTime() < expirationTime) {
        const username = localStorage.getItem('currentUser');
        showLoggedInUser(username);
    }
});

// Відображення імені користувача та кнопки виходу
function showLoggedInUser(username) {
    const loginInfo = document.getElementById('login-info');
    loginInfo.innerHTML = `Ви увійшли як: ${username} | <a href="#"
onclick="logout()">Вихід</a>`;
    loginInfo.style.display = 'block';
}

// Функція для виходу з облікового запису
function logout() {

```

```
localStorage.removeItem('currentUser');
localStorage.removeItem('currentRole');
localStorage.removeItem('expirationTime');
location.reload(); // Перезавантажити сторінку після виходу
}

/* style.css */
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    text-align: center;
}

header {
    background: #4CAF50;
    color: white;
    padding: 10px 0;
}

nav ul {
    list-style-type: none;
    padding: 0;
    background: #333;
}

nav ul li {
    display: inline;
    margin-right: 10px;
}
```

```
nav ul li a {  
    text-decoration: none;  
    color: white;  
    padding: 15px 20px;  
    display: inline-block;  
}
```

```
nav ul li a:hover {  
    background: #4CAF50;  
}
```

```
nav ul li a.active {  
    background: #66ff66;  
    border-radius: 10px;  
}
```

```
main {  
    padding: 20px;  
    text-align: left;  
}
```

```
.content-section {  
    display: none;  
    text-align: center;  
}
```

```
.large-text {  
    font-size: 2em;  
    text-align: center;
```

```
}
```

```
#map {  
  width: 100%;  
  height: 800px;  
  display: block;  
  margin: 20px auto;  
}
```

```
.parking-grid {  
  display: grid;  
  grid-template-columns: repeat(10, 50px);  
  gap: 10px;  
  justify-content: center;  
}
```

```
.parking-grid.large-gap {  
  gap: 20px;  
}
```

```
.parking-space {  
  width: 50px;  
  height: 50px;  
  background: #ccc;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  cursor: pointer;  
  border: 1px solid #999;  
}
```

```
.parking-space.occupied {  
    background: #ff6666;  
}
```

```
.parking-space.free {  
    background: #66ff66;  
}
```

```
.parking-image {  
    display: block;  
    margin: 20px auto;  
    max-width: 100%;  
    height: auto;  
}
```

```
footer {  
    background: #333;  
    color: white;  
    text-align: center;  
    padding: 10px 0;  
    position: fixed;  
    bottom: 0;  
    width: 100%;  
}
```

```
/* Додаємо стилі для форм авторизації та реєстрації */
```

```
form {  
    display: flex;  
    flex-direction: column;
```

```
max-width: 300px;
margin: 0 auto;
text-align: center; /* Центруємо форми */
}

form label, form input, form button {
margin-bottom: 10px;
text-align: center; /* Центруємо елементи форм */
}

#registrationForm {
border: 1px solid #ccc;
padding: 20px;
border-radius: 5px;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
background-color: #f9f9f9;
}

/* Стили для форм авторизації та реєстрації */
form {
display: flex;
flex-direction: column;
max-width: 300px;
margin: 0 auto;
}

form label, form input, form button {
margin-bottom: 10px;
}
```

```
/* Стили для секцій авторизації та реєстрації */
#login, #registration {
    text-align: center;
}

#loginForm, #registrationForm {
    border: 1px solid #ccc;
    padding: 20px;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    background-color: #f9f9f9;
}

#admin-indicator {
    position: fixed;
    top: 10px;
    right: 10px;
    z-index: 9999;
}

body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f0f0f0; /* Колір фону сторінки */
}

main {
    padding: 20px;
    background-color: #fff; /* Колір фону контейнера вмісту */
}
```

```
border-radius: 10px; /* Заокруглення кутів контейнера */  
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1); /* Тінь контейнера */  
}
```

```
button, .book-button {  
  padding: 10px 20px;  
  font-size: 16px;  
  border: none;  
  border-radius: 5px;  
  cursor: pointer;  
  transition: background-color 0.3s ease;  
}
```

```
button:hover, .book-button:hover {  
  background-color: #4CAF50;  
  color: #fff;  
}
```

```
.about-content {  
  text-align: center; /* Центруємо вміст */  
}
```

```
.about-content img {  
  display: block;  
  margin: 20px auto; /* Центруємо зображення */  
}
```

```
.section-title {  
  text-align: center; /* Центруємо заголовок */  
}
```

```
.temporary-reserved {  
    background-color: red !important;  
}
```

Arduino скетч:

```
#include <ESP8266WiFi.h>  
#include <WiFiClient.h>  
  
const char* ssid = "VIKI"; // SSID Wi-Fi мережі  
const char* password = "88888888"; // Пароль Wi-Fi мережі  
  
WiFiServer server(80);  
  
const int trigPin = 9;  
const int echoPin = 10;  
  
void setup() {  
    Serial.begin(115200);  
    pinMode(trigPin, OUTPUT);  
    pinMode(echoPin, INPUT);  
  
    // Підключення до Wi-Fi  
    Serial.println();  
    Serial.println();  
    Serial.print("Підключення до ");  
    Serial.println(ssid);  
  
    WiFi.begin(ssid, password);
```

```
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}

Serial.println("");
Serial.println("WiFi підключено");
Serial.println("IP адреса: ");
Serial.println(WiFi.localIP());

server.begin();
}

void loop() {
  long duration;
  int distance;

  // Вимірювання відстані з датчика HC-SR04
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2; // Розрахунок відстані в см

  // Перевірка зайнятості місця
  int status = (distance > 400) ? 1 : 0;
```

```
// Надсилання даних по Wi-Fi
WiFiClient client = server.available();
if (client) {
  Serial.println("Новий клієнт.");
  String request = client.readStringUntil('\r');
  Serial.println(request);
  client.flush();

  String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n";
  s += "Parking status: ";
  s += status;
  client.print(s);
  delay(1);
  Serial.println("Клієнт відключився.");
}
delay(2000); // Очікування перед наступним вимірюванням
}
```

Прийом і відображення даних з датчиків

```
async function fetchParkingStatus() {
  try {
    const response = await fetch('http://<ESP8266_IP_ADDRESS>');
    const data = await response.text();
    const status = parseInt(data.match(/Parking status: (\d)/)[1], 10);
    updateSensorData(status);
  } catch (error) {
    console.error('Помилка при отриманні даних з ESP8266:', error);
  }
}
```

```
function updateSensorData(status) {  
  // Оновлення стану паркувального місця  
  const space = document.querySelector('.parking-space');  
  space.className = `parking-space ${status ? 'free' : 'occupied'}`;  
}  
  
setInterval(fetchParkingStatus, 5000); // Оновлення кожні 5 секунд
```

ДОДАТОК Б**Демонстраційні матеріали у вигляді презентації**

Харківський національний університет радіоелектроніки
Факультет Автоматики і комп'ютеризованих технологій
Кафедра КІТАР

Спеціальність 151 Автоматизація та комп'ютерно інтегровані технології
Кваліфікаційна робота на тему:

**«Розроблення системи автоматизації для
оптимізації паркування автотранспорту на базі
технології IoT»**

Виконав:
студент групи АКТАКІТ-20-2
Жердев М. В.

Керівник роботи:
доц. каф. КІТАР
Хрустальова С. В.

Мета і завдання роботи

Мета роботи – підвищення ефективності управління паркувальними місцями на підземній парковці.

Об'єкт розробки – процес моніторингу та бронювання паркувальних місць у програмному виді.

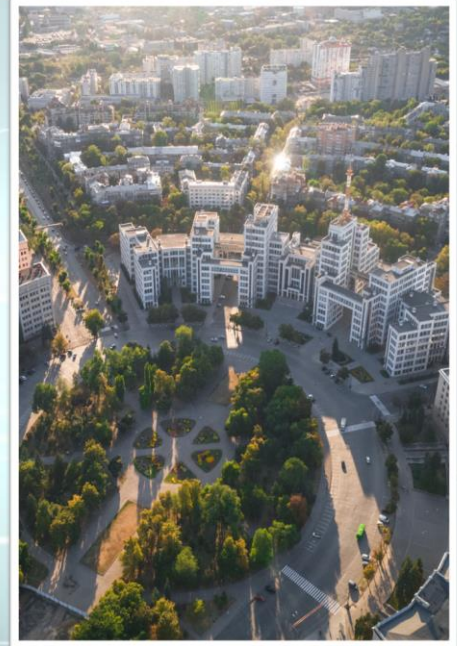
Предмет розробки – програмний засіб для моніторингу та бронювання паркувальних місць.

Завдання:

- провести огляд та аналіз існуючих методів, засобів та систем оптимізації паркування автотранспорту;
- розробити структурну схему системи;
- обрати компоненти для реалізації системи;
- обрати програмне забезпечення;
- реалізувати систему у вигляді програмного засобу.

Актуальність роботи

Сучасні міста стикаються з проблемою недостатньої кількості паркувальних місць. Автоматизація паркування з використанням технології IoT дозволяє оптимізувати використання паркувальних зон, знижуючи витрати та покращуючи зручність для водіїв.



Аналіз існуючих систем

Проаналізовані системи:

- Park System
- AFAPARK
- FidPark

Найефективнішими є ті системи, які використовують технології ІоТ для моніторингу паркувальних місць. Це дозволяє забезпечити точне визначення наявності автомобіля на місці та своєчасну передачу даних до центральної системи управління.



Структурна схема системи автоматизації паркування автотранспорту

Система автоматизації паркування включає можливості моніторингу зайнятості паркувальних місць, передачі даних у режимі реального часу на сервер, а також відображення інформації для користувачів.



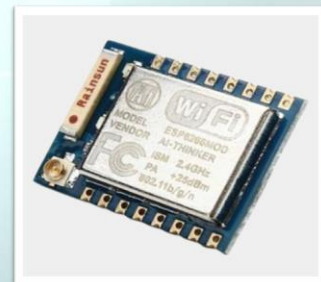
Вибір компонентів системи

Для створення системи автоматизації паркування було обрано такі основні компоненти:

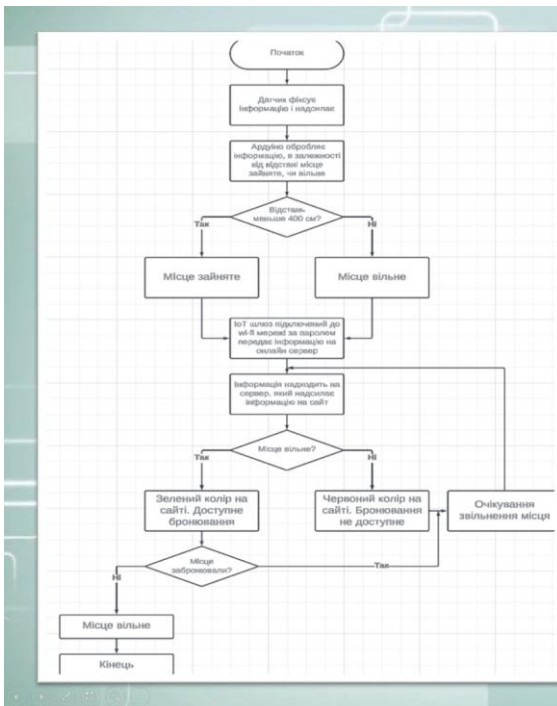
- а) ультразвуковий датчик HC-SR04
 - використовуються для визначення наявності автомобіля на паркувальному місці.
 - висока точність вимірювань відстані.
- б) контролер Arduino Uno
 - центральний контролер для збору та обробки даних від датчиків.
 - легко програмується та має достатню кількість входів/виходів.
- в) Wi-Fi модуль ESP8266
 - забезпечує передачу даних від контролера веб-сайт через Wi-Fi.



Ультразвуковий датчик відстані HC-SR04



Мікроконтролерний модуль ESP8266



Алгоритм роботи системи автоматизації паркування автотранспорту

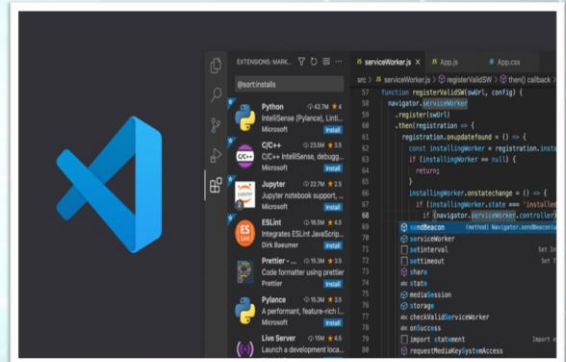
- початок;
- ультразвуковий датчик фіксує наявність або відсутність автомобіля на паркувальному місці;
- контролер Arduino обробляє дані від датчика та визначає, чи зайняте місце;
- якщо відстань менше 400 см, місце зайняте; інакше місце вільне;
- дані передаються через Wi-Fi модуль ESP8266 на сервер;
- сервер оновлює інформацію про стан місць на веб-сайті;
- на сайті відображається зелений колір, якщо місце вільне, або червоний, якщо зайняте;
- якщо місце вільне, користувач може забронювати його через сайт, або зайняти його;
- якщо місце заброньоване, воно відображається як зайняте і очікує час бронювання, після якого датчик оновлює інформацію про стан місця;
- кінець.

Вибір програмного забезпечення і реалізації системи у вигляді програмного засобу

Мови програмування:

- HTML використовується для створення структури веб-сторінок та інтерфейсу користувача.
- C++ використовується для поліпшення вигляду інтерфейсу .
- JavaScript використовується для взаємодії з окремими частинами інтерфейсу (Наприклад, перехід між сторінками, або бронювання паркувальних місць)

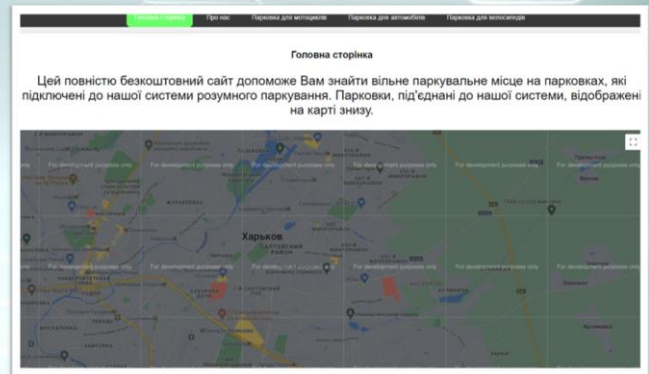
Середовище розробки: Visual Studio Code.



Програмний засіб

Розроблений програмний засіб надає користувачам ряд можливостей:

- Авторизація та реєстрація користувачів для доступу до системи.
- Відображення доступних паркувальних місць у реальному часі.
- Інформаційний розділ із загальною інформацією
- Моніторинг наявності вільних місць на парковці
- Система бронювання паркувальних місць, що дозволяє зарезервувати місце на певний час.



Kharkiv Smart Parking

Авторизація

Парковка для автомобілів

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40

Номер місця:

Час бронювання (хв):

Висновки

- У процесі виконання кваліфікаційної роботи було проведено аналіз існуючих систем оптимізації паркування автотранспорту, зокрема їх технічних характеристик та способів реалізації.
- Розроблено загальну структурну схему системи автоматизації для оптимізації паркування та алгоритм її роботи
- Обрані компоненти для реалізації системи автоматизації для оптимізації паркування
- Створено програмну частину, яка включає розробку веб-додатка для моніторингу та бронювання паркувальних місць у реальному часі.
- Проведено тестування програмного забезпечення для забезпечення його надійної роботи та визначення можливих проблем.

Таким чином, в результаті виконання кваліфікаційної роботи, підвищено ефективність управління паркувальними місцями на підземній парковці.

Також, отримані результати роботи можна віднести до Цілі сталого розвитку 9 «Промисловість, інновації та інфраструктура», а саме п.9.3 «Забезпечити доступність дорожньотранспортної інфраструктури, яка базується на використанні інноваційних технологій, зокрема через розширення форм участі держави у різних інфраструктурних проектах».

Питання та відповіді

Дякую за увагу!

Я готовий відповісти на ваші питання щодо проекту, його реалізації та можливостей подальшого розвитку.



