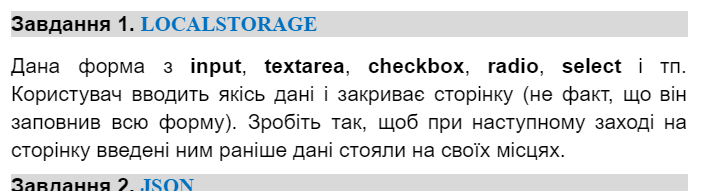
**Лабораторна робота №10**

**Тема:** LocalStorage. JSON. Наслідування класу. Статичні властивості та методи

***Мета роботи:*** вивчити засоби по роботі з cookie, localStorage, JSON, освоїти роботу з класами в JavaScript

**Хід роботи:**

***Завдання 1***.



Лістинг програми:

Index.html

<form action="/" method="post">

      <div>

        <label for="name">Ім'я:</label>

        <input type="text" id="name" name="name" placeholder="Ім'я">

      </div>

      <div>

        <label for="surname">Прізвище:</label>

        <input type="text" id="surname" name="surname" placeholder="Прізвище">

      </div>

      <div class="radio-group">

        <label>

          <input type="radio" name="gender" value="male"> Чоловік

        </label>

        <label>

          <input type="radio" name="gender" value="female"> Жінка

        </label>

      </div>

      <div>

        <label for="country">Країна:</label>

        <select id="country" name="country">

          <option value="ukraine">Україна</option>

          <option value="china">Китай</option>

          <option value="usa">США</option>

        </select>

      </div>

      <div>

        <label for="comment">Коментар:</label>

        <textarea id="comment" name="comment" placeholder="Введіть свій коментар"></textarea>

      </div>

      <button type="submit">Відправити</button>

    </form>

Style.css

body {

   font-family: Arial, sans-serif;

   background-color: #f4f4f4;

   margin: 0;

   display: flex;

   justify-content: center;

   align-items: center;

   height: 100vh;

 }

 form {

   background-color: #fff;

   padding: 20px;

   border-radius: 8px;

   box-shadow: 0 0 10px *rgba*(0, 0, 0, 0.1);

 }

 input,

 select,

 textarea {

   width: 100%;

   padding: 10px;

   margin: 8px 0;

   box-sizing: border-box;

 }

 label {

   display: block;

   margin-bottom: 8px;

 }

 .checkbox-group,

 .radio-group {

   display: flex;

   flex-direction: column;

 }

 .checkbox-group label,

 .radio-group label {

   display: flex;

   align-items: center;

 }

 input[type="checkbox"],

 input[type="radio"] {

   margin-right: 5px;

 }

 button {

   background-color: #4caf50;

   color: #fff;

   padding: 10px;

   border: none;

   border-radius: 4px;

   cursor: pointer;

 }

 button:hover {

   background-color: #45a049;

 }

Script.js

*const* form = document.querySelector("form");

*// Зберігати дані форми в локальному сховищі при введенні користувачем*

form.addEventListener("input", () *=>* {

*const* formData = *new* FormData(form);

*for* (*const* [key, value] of formData) {

*if* (value.trim() !== '') {

      localStorage.setItem(key, value);

    }

  }

});

*// Відновлення введених раніше даних при завантаженні сторінки*

window.onload = () *=>* {

*for* (*const* input of document.querySelectorAll("input, textarea, select")) {

*const* name = input.name;

*const* value = localStorage.getItem(name);

*if* (value !== null) {

*if* (input.type === 'radio') {

        input.checked = (input.value === value);

      } *else* {

        input.value = value;

      }

    }

  }

};

*// Блокування відправки форми при натисканні кнопки "Відправити"*

*const* submitButton = document.querySelector("button[type='submit']");

*if* (submitButton) {

  submitButton.addEventListener("click", (event) *=>* {

    event.preventDefault();

  });

}

console.log(localStorage);

Результати виконання:

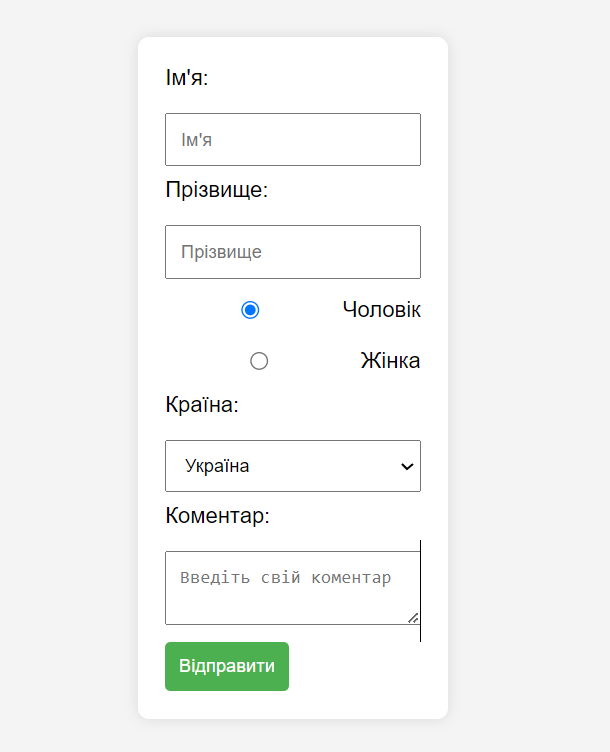


Рисунок 1. Перше завантаження

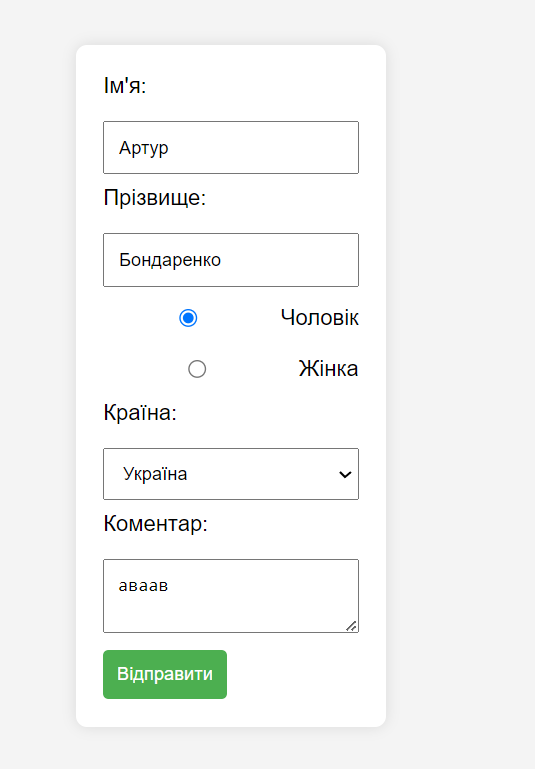


Рисунок 2. Заповненна форма

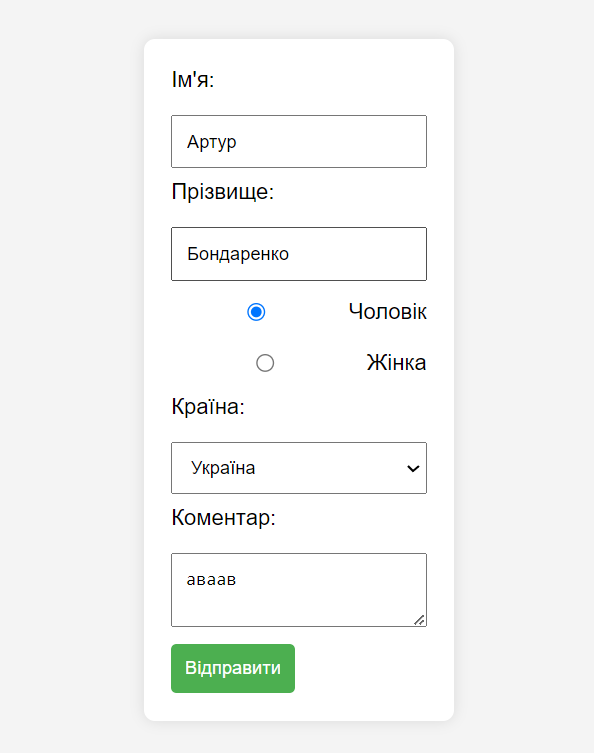


Рисунок 3. Відкрита форма у новому вікні браузера(перевіряємо ЛокалСторедж)

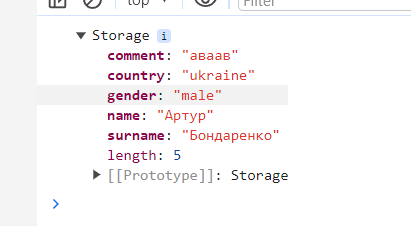
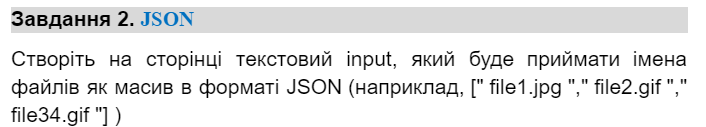


Рисунок 4. Ключі та значення ЛокалСторедж

***Завдання 2***.



Лістинг програми:

Index.html

<div class="container">

      <div class="header">

        <input type="text" id="fileInput" placeholder="Введіть масив ">

        <button id="showImagesButton">Show image</button>

      </div>

      <div id="imageContainer"></div>

      <span id="message"></span>

      <div id="error"></div>

    </div>

Style.css

.container {

   max-width: 800px;

   margin: 50px auto;

   text-align: left;

 }

 .input {

   max-width: *calc*(100%);

 }

 button {

   margin-left: 10px;

 }

 img {

   max-width: 200px;

   max-height: 150px;

   margin: 10px;

   cursor: pointer;

   transition: transform 0.3s, box-shadow 0.3s;

 }

 img:hover {

   box-shadow: 0 0 10px *rgba*(0, 0, 0, 0.5);

 }

 #error {

   color: red;

 }

 #imageContainer {

   margin-top: 150px;

 }

 .image-active {

   position: relative;

   z-index: 1;

   transform: *scale*(1);

 }

 .image-effects {

   position: relative;

   z-index: 100;

   transform: *scale*(3);

 }

Script.js

*function* showImages() {

*const* fileInput = document.getElementById('fileInput');

*const* imageContainer = document.getElementById('imageContainer');

*const* message = document.getElementById('message');

*const* error = document.getElementById('error');

*try* {

*const* files = JSON.parse(fileInput.value);

*if* (!Array.isArray(files)) {

*throw* *new* Error('не вірне значення');

     }

     imageContainer.innerHTML = '';

     files.forEach((file, index) *=>* {

*const* imgElement = document.createElement('img');

       imgElement.src = file.trim();

       imgElement.alt = 'Image';

       imgElement.classList.add('image-active');

       imageContainer.appendChild(imgElement);

       imgElement.addEventListener('click', () *=>* {

         imgElement.classList.toggle('image-effects');

       });

     });

     message.textContent = 'Успішно завантажено';

     message.style.position = 'fixed';

     message.style.top = '10px';

     message.style.right = '10px';

     message.style.color = 'green';

     message.style.zIndex = '1000';

     error.textContent = '';

   } *catch* (err) {

     message.textContent = '';

     error.textContent = err.message;

   }

 }

 window.onload = () *=>* {

   showImages();

*const* showImagesButton = document.getElementById('showImagesButton');

   showImagesButton.addEventListener('click', showImages);

 };

*// ["car1.jpeg","car2.jpg","car3.jpeg"]*

Результати виконання:

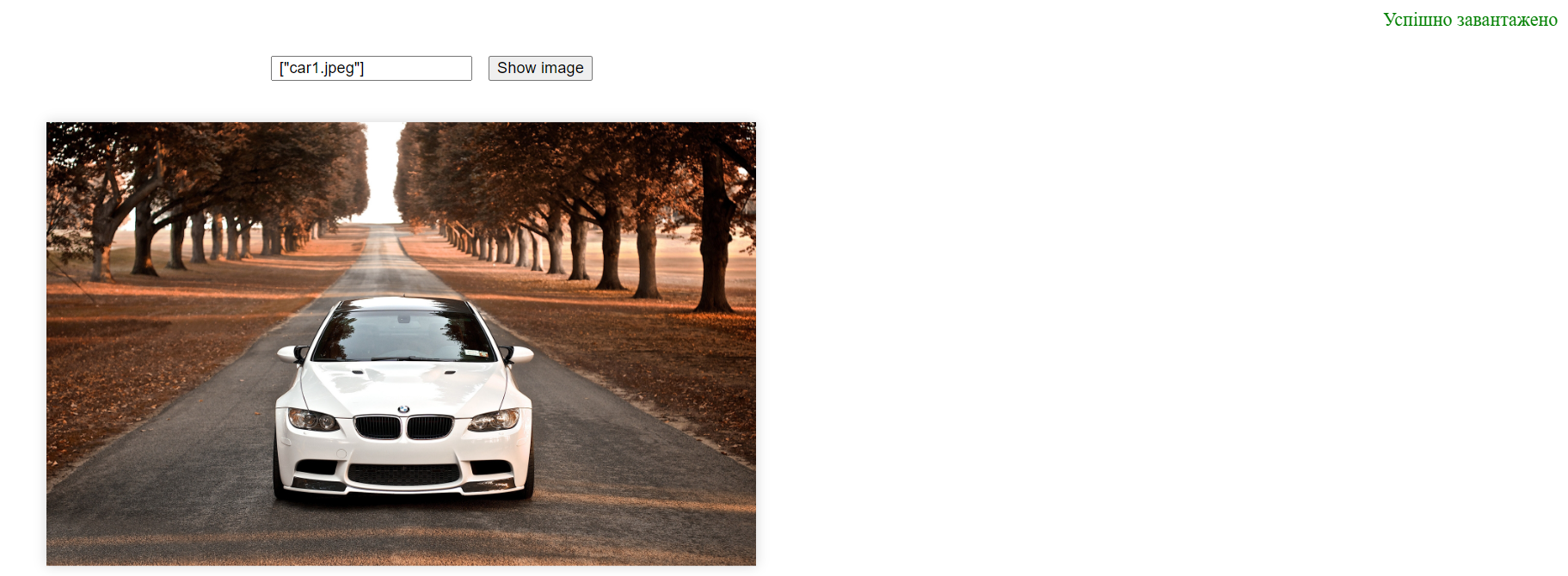


Рисунок 5. Форма вводу JSON масиву



Рисунок 6. Виведений елемент масиву

У правому верхньому кутку повідомлення про успішне вивантаження масиву



Розмір 7. По натисканню змінюємо розмір

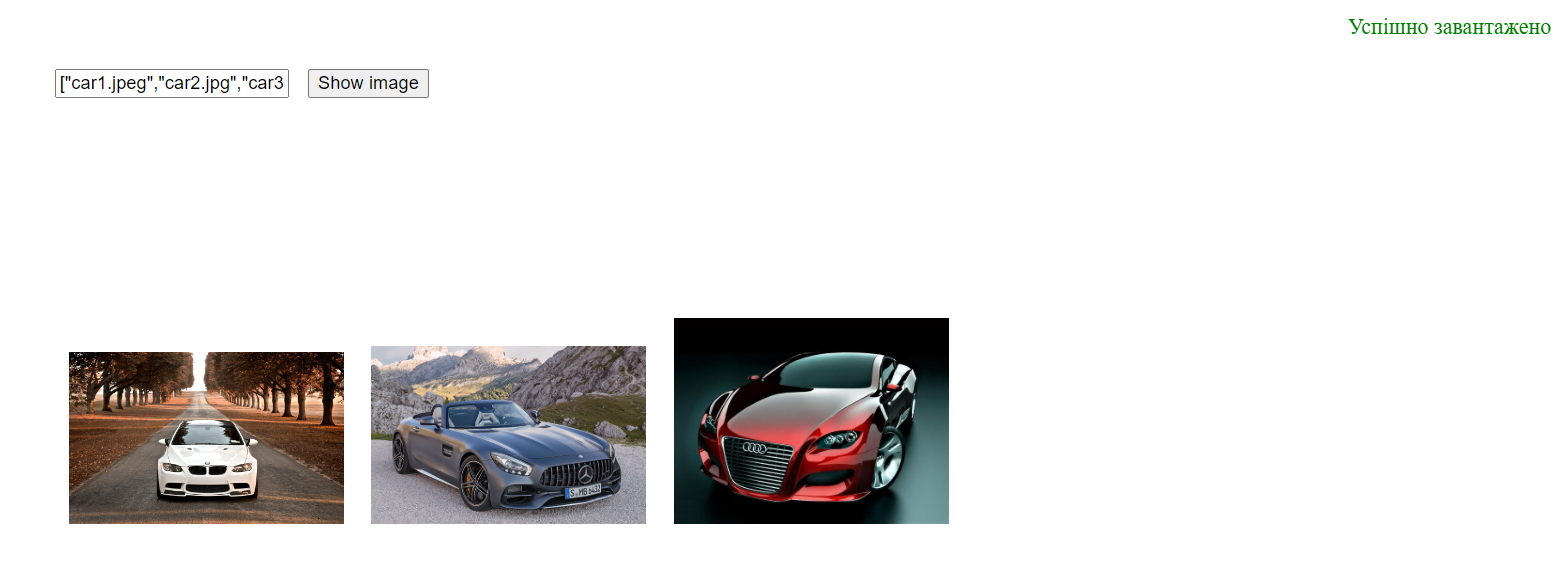


Рисунок 8. Виводимо 3 елементи масиву

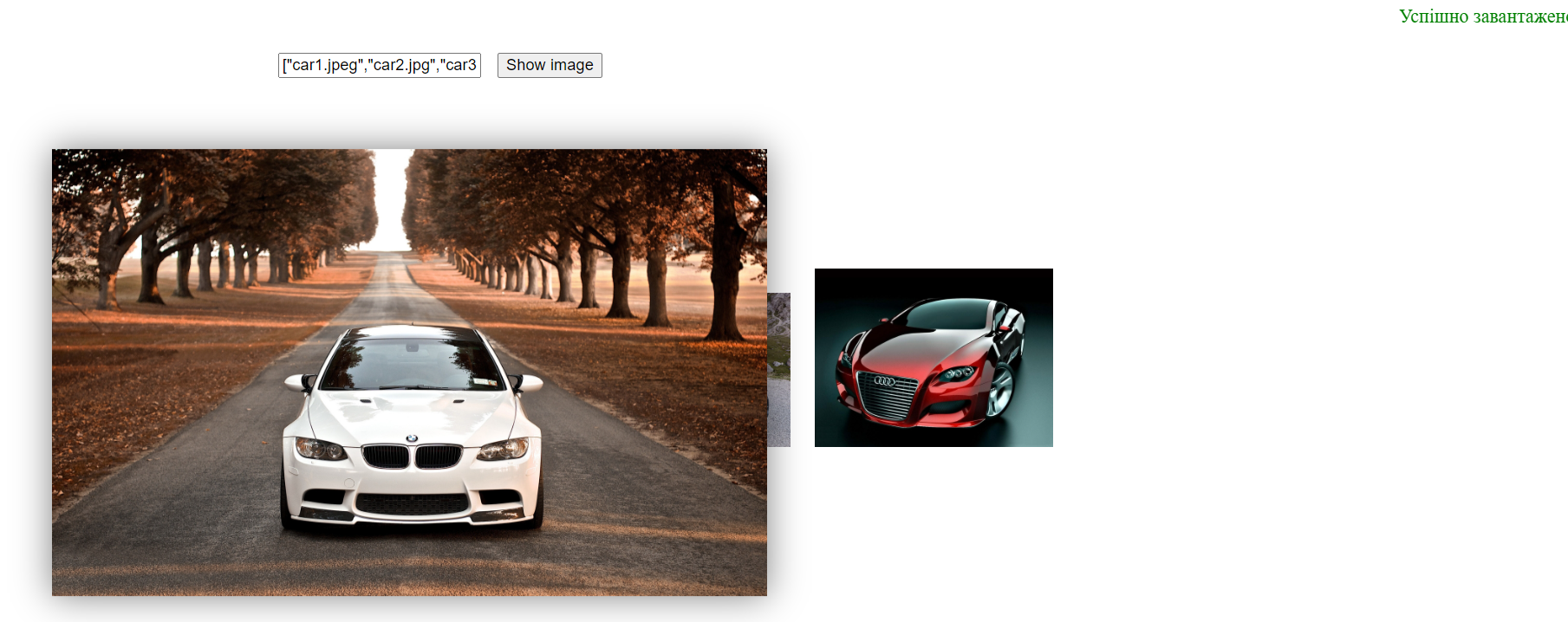


Рисунок 9. Виконане завдання

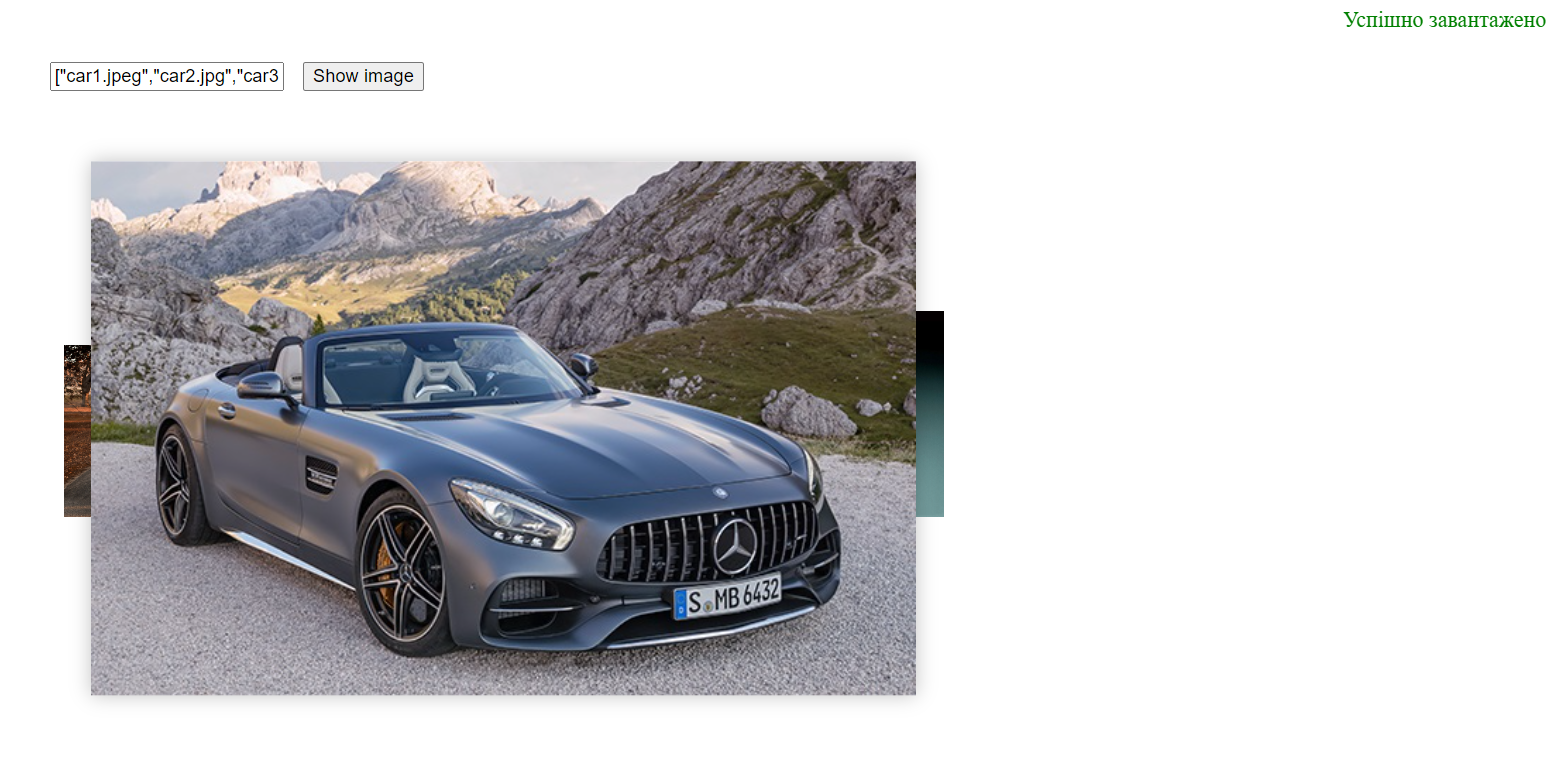


Рисунок 10. Виконане завдання

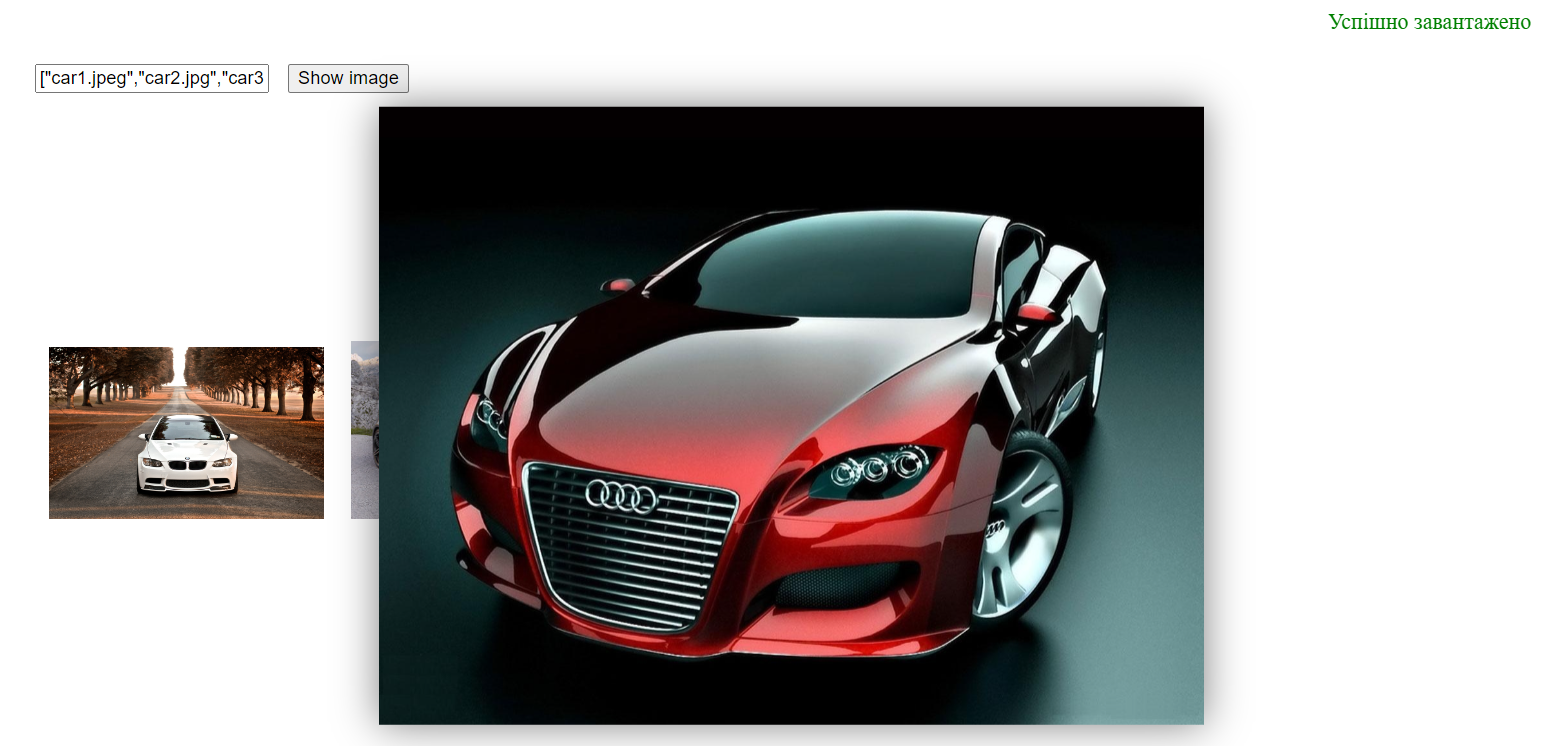
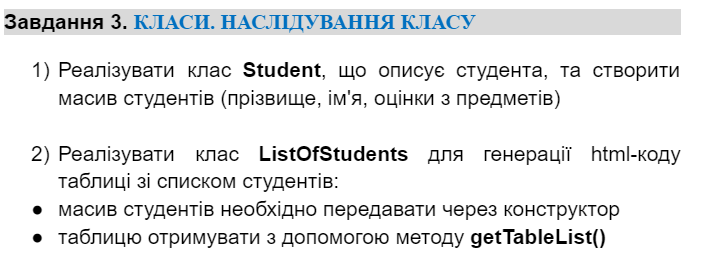


Рисунок 11. Виконане завдання

***Завдання 3***.



Лістинг програми:

Index.html

<div id="listOfStudents"></div>

<div id="styledTable"></div>

Style.css

table {

   width: 50%;

}

th, td {

   border: 3px solid black;

   padding: 8px;

   text-align: left;

}

tbody{

   color:  black;

}

thead {

   background-color: lightgreen;

   color: white !important;

}

Script.js

*class* Student {

*constructor*(lastName, firstName, math, history, JS, avg) {

*this*.lastName = lastName;

*this*.firstName = firstName;

*this*.math = math;

*this*.history = history;

*this*.JS = JS;

*this*.avg = avg;

   }

 }

*class* StudentList {

*constructor*(students) {

*this*.students = students;

   }

   generateTable() {

*let* table = '<table border="1"><thead><tr><th>Last Name</th><th>First Name</th><th>Math</th><th>History</th><th>JS</th><th>Avg</th></tr></thead>';

*for* (*let* student of *this*.students) {

       table += `<tbody><tr><td>${student.lastName}</td><td>${student.firstName}</td><td>${student.math}</td><td>${student.history}</td><td>${student.JS}</td><td>${student.avg}</td></tr></tbody>`;

     }

     table += '</table>';

     table += `<br><span class="green-text">Загальний середній бал по групі: ${*this*.calculateTotalAvg()}</span><br>`;

*return* table;

   }

   calculateTotalAvg() {

*let* totalAvg = *this*.students.reduce((acc, student) *=>* acc + Number(student.avg), 0);

*return* (totalAvg / *this*.students.length).toFixed(2);

   }

 }

*class* StyledStudentTable *extends* StudentList {

   getStyles() {

*return* 'table { border-collapse: collapse; width: 50%; } th, td { border: 1px solid black; padding: 8px; text-align: left; }';

   }

   generateStyledTable() {

*const* parentTable = *super*.generateTable();

*const* styles = *this*.getStyles();

*return* `<style>${styles}</style>${parentTable}`;

   }

   calculateAverageForStudents() {

*for* (*let* student of *this*.students) {

       student.avg = ((student.math + student.history + student.JS) / 3).toFixed(2);

     }

   }

 }

*const* studentsData = [

*new* Student('Артур', 'Лиховида', 2, 3, 3),

*new* Student('Вадим', 'Бандура', 3, 5, 4),

*new* Student('Олексій', 'Олав', 5, 2, 4),

*new* Student('Олексій', 'Олав', 5, 5, 5),

*new* Student('Олексій', 'Олав', 5, 3, 3),

*new* Student('Олексій', 'Олав', 5, 1, 1),

*new* Student('Олексій', 'Олав', 5, 5, 5),

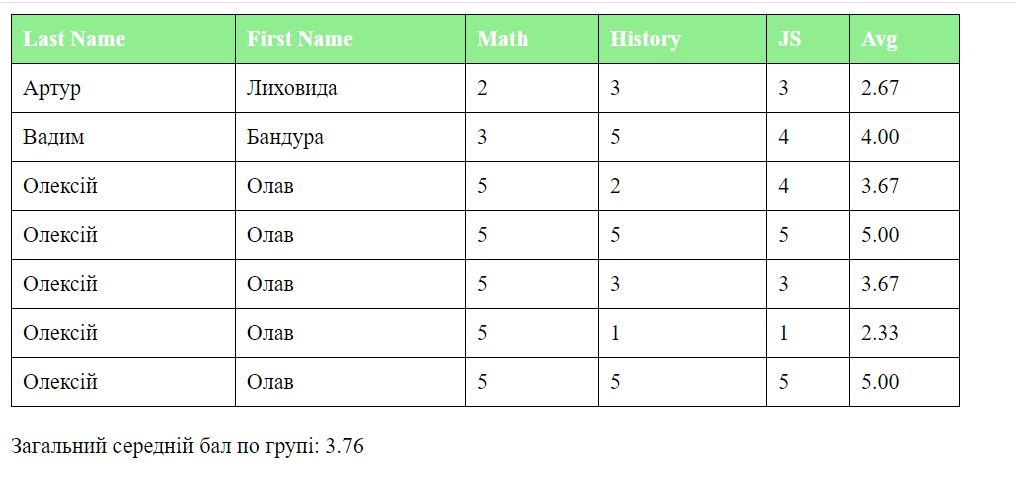
 ];

*const* styledStudentTable = *new* StyledStudentTable(studentsData);

 styledStudentTable.calculateAverageForStudents();

 document.getElementById('styledTable').innerHTML = styledStudentTable.generateStyledTable();

Результати виконання:

Рисунок 12. Виконане завдання, відмальована таблиця

***Завдання 4***.



Лістинг програми:

Index.html

   <button id="draw">draw</button>

  <button id="fill">fill</button>

  <div id="container"></div>

Style.css

#container {

   display: flex;

   margin-top: 15px;

 }

 .shape {

   width: 30px;

   height: 30px;

   border: 1px solid;

 }

Script.js

*class* Shape {

*static* total = 100;

*constructor*(a) {

*this*.a = a;

   }

*static* fill() {

     Shape.total = 100;

   }

   draw() {

*let* quad = document.createElement("div");

     quad.classList.add("shape");

     quad.style.backgroundColor = `rgba(255, 0, 0, ${Shape.total / 100})`;

     quad.style.borderColor = "black";

     Shape.total -= 10;

*return* quad;

   }

 }

*let* btnDraw = document.querySelector("#draw");

*let* btnFill = document.querySelector("#fill");

*let* container = document.querySelector("#container");

 btnDraw.addEventListener("click", *function* () {

*let* s = *new* Shape(100);

   container.appendChild(s.draw());

 });

 btnFill.addEventListener("click", *function* () {

   Shape.fill();

 });

Результати виконання:



Рисунок 13. Кнопки малювання

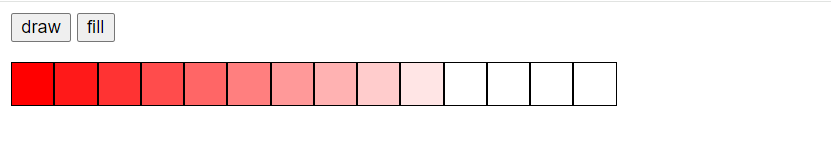


Рисунок 14. Процес втрачання кольору

Під час кожного відмальованого квадратика, втрачається насиченість кольором на 10%. Десятий блок вже буде абсолютно прозорим

Натискаємо кнопку **FILL, тобто заповнюємо варбую та малюємо далі**

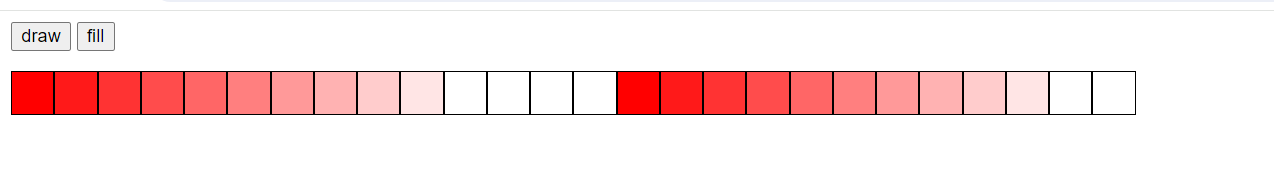


Рисунок 15. Процес оновлення кольору і малювання знов