**Лабораторна робота №6**

**Тема**: Події.

**Мета роботи**: вивчити засоби по роботі з об’єктною моделлю документа; навчитися додавати, змінювати та видаляти елементи веб-сторінки; засвоїти принципи роботи з CSS-стилями

**Завдання 1.**

Дана таблиця з користувачами: номер, ім'я та прізвище. Під таблицею зробіть форму, з допомогою якої можна буде додати нового користувача в таблицю.

**Лістинг коду**

**HTML:**

<table id="myTable">

        <tr>

            <td>

                <div class="number">№</div>

            </td>

            <td>

                <div class="FirstName">FirstName</div>

            </td>

            <td>

                <div class="LastName">LastName</div>

            </td>

        </tr>

    </table>

    <input type="text" id="addPersonFirstName">

    <input type="text" id="addPersonLastName">

    <button id="checkButton">Додати</button>

**JavaScript:**

let count = 0;

        let myTable = document.getElementById("myTable");

        checkButton.addEventListener("click", function () {

            let addPersonFirstName = document.getElementById("addPersonFirstName");

            let addPersonLastName = document.getElementById("addPersonLastName");

            let newPersonFirstName = addPersonFirstName.value;

            let newPersonLastName = addPersonLastName.value;

            let tr = document.createElement("tr");

            let tdNumber = document.createElement("td");

            tdNumber.innerHTML = "<div>" + count + "</div>";

            tr.appendChild(tdNumber);

            let tdFirstName = document.createElement("td");

            tdFirstName.innerHTML = "<div>" + newPersonFirstName + "</div>";

            tr.appendChild(tdFirstName);

            let tdLastName = document.createElement("td");

            tdLastName.innerHTML = "<div>" + newPersonLastName + "</div>";

            tr.appendChild(tdLastName);

            count++;

            myTable.appendChild(tr);

        });

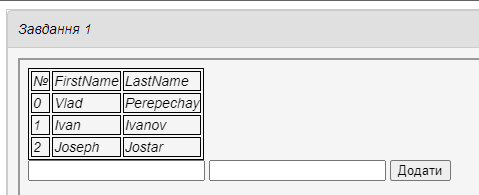


Рис 6.1

**Завдання 2.**

В елемент textarea вводяться нулі та одиниці і з них потрібно сформувати квадратики чорного і білого кольору.

**Лістинг коду**

**HMTL:**

<textarea id="binaryInput" rows="5" cols="30"></textarea>

    <button onclick="generateSquares()">Генерувати квадратики</button>

    <div class="container1" id="squaresContainer"></div>

**JavaScript:**

function generateSquares() {

            const textarea = document.getElementById('binaryInput');

            const squaresContainer = document.getElementById('squaresContainer');

            squaresContainer.innerHTML = '';

            const lines = textarea.value.split('\n');

            lines.forEach(line => {

                const row = document.createElement('div');

                row.classList.add('container');

                let newline = document.createElement("br");

                for (let i = 0; i < line.length; i++) {

                    const square = document.createElement('div');

                    square.classList.add('square');

                    square.style.backgroundColor = line[i] === '1' ? 'black' : 'white';

                    row.appendChild(square);

                }

                squaresContainer.appendChild(row);

            });

        }



Рис 6.2

**Завдання 3.**

Вибрати потрібні кольори з палітри кольорів та залити ними даний прямокутник (блок). Кольори в палітрі повинні вибиратись клацанням. Повторне клацання повинно відміняти вибір. Заливка прямокутника відбувається одночасно з вибором кольорів. Якщо вибрано декілька кольорів, то заливка градієнтна.

**Лістинг коду**

**HTML:**

<button id="checkButton">Додати</button>

    <table id="myTable">

        <tr>

            <td style="background-color: red;" id="colorRed" data-active="false" ondblclick="doubleClick(this)"></td>

            <td style="background-color: blue;" id="colorBlue" data-active="false" ondblclick="doubleClick(this)"></td>

            <td style="background-color: yellow;" id="colorYellow" data-active="false" ondblclick="doubleClick(this)">

            </td>

        </tr>

        <tr>

            <td style="background-color: green;" id="colorGreen" data-active="false" ondblclick="doubleClick(this)">

            </td>

            <td style="background-color: black;" id="colorBlack" data-active="false" ondblclick="doubleClick(this)">

            </td>

            <td style="background-color: white;" id="colorWhite" data-active="false" ondblclick="doubleClick(this)">

            </td>

        </tr>

        <tr>

            <td style="background-color: pink;" id="colorPink" data-active="false" ondblclick="doubleClick(this)"></td>

            <td style="background-color: purple;" id="colorPurple" data-active="false" ondblclick="doubleClick(this)">

            </td>

            <td style="background-color: magenta;" id="colorMagenta" data-active="false" ondblclick="doubleClick(this)">

            </td>

        </tr>

    </table>

    <canvas id="gradient"></canvas>

**JavaScript:**

 function doubleClick(color) {

            const isActive = color.getAttribute("data-active");

            if (isActive === "false") {

                color.style.border = "4px solid lightblue";

                color.setAttribute("data-active", "true");

            } else {

                color.style.border = "0px";

                color.setAttribute("data-active", "false");

            }

        }

        let checkButton = document.querySelector("#checkButton");

        color = document.querySelector("#inputColor");

        checkButton.addEventListener("click", function () {

            let arrColors = [];

            let allColors = document.querySelectorAll("td");

            let rectangleGradient = document.querySelector("#gradient");

            for (let i = 0; i < allColors.length; i++) {

                switch (allColors[i].id) {

                    case "colorRed": {

                        if (allColors[i].getAttribute("data-active") === "true") {

                            arrColors.push("red");

                        }

                    }

                        break;

                    case "colorBlue": {

                        if (allColors[i].getAttribute("data-active") === "true") {

                            arrColors.push("blue");

                        }

                    }

                        break;

                    case "colorYellow": {

                        if (allColors[i].getAttribute("data-active") === "true") {

                            arrColors.push("yellow");

                        }

                    }

                        break;

                    case "colorGreen": {

                        if (allColors[i].getAttribute("data-active") === "true") {

                            arrColors.push("green");

                        }

                    }

                        break;

                    case "colorBlack": {

                        if (allColors[i].getAttribute("data-active") === "true") {

                            arrColors.push("black");

                        }

                    }

                        break;

                    case "colorWhite": {

                        if (allColors[i].getAttribute("data-active") === "true") {

                            arrColors.push("white");

                        }

                    }

                        break;

                    case "colorPink": {

                        if (allColors[i].getAttribute("data-active") === "true") {

                            arrColors.push("pink");

                        }

                    }

                        break;

                    case "colorPurple": {

                        if (allColors[i].getAttribute("data-active") === "true") {

                            arrColors.push("purple");

                        }

                    }

                        break;

                    case "colorMagenta": {

                        if (allColors[i].getAttribute("data-active") === "true") {

                            arrColors.push("magenta");

                        }

                    }

                        break;

                }

            }

            console.log(arrColors);

            let ctx = rectangleGradient.getContext("2d");

            let gradient = ctx.createLinearGradient(0, 0, 170, 0);

            let j = 0.1;

            for (let i = 0; i < arrColors.length; i++) {

                gradient.addColorStop(j, arrColors[i]);

                j += 0.1;

            }

            ctx.fillStyle = gradient;

            ctx.fillRect(20, 20, 150, 100);

        });

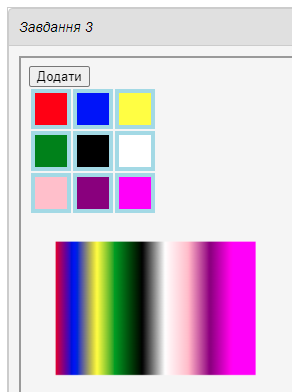


Рис 6.3

**Завдання 4.**

Є блоки з текстом. В кожному з блоків є кнопки Видалити та Редагувати

При натисканні на кнопку Видалити, даний блок зникає. При натисканні на Редагувати блок переходить в стан редагування (наприклад, з допомогою textarea). При знятті фокуса з textarea, блок повертається в звичайний стан.

**Лістинг коду**

**HTML:**

<button id="addBlock">Додати поле</button>

    <div id="mainBlock"></div>

**JavaScript:**

let countBlocks = 0;

        let addBlock = document.querySelector("#addBlock");

        let content = "Lorem ipsum is placeholder text commonly used in the graphic, print, and publishing industries";

        addBlock.addEventListener("click", function () {

            let mainBlock = document.querySelector("#mainBlock");

            let block = document.createElement("div");

            block.classList.add("block");

            countBlocks++;

            block.innerHTML += "<button id=\"buttonEdit\" onclick=\"Edit(this)\">Редагувати</button>";

            block.innerHTML += "<button id=\"buttonDelete\" onclick=\"Delete(this)\">Видалить</button>";

            block.innerHTML += "<br>";

            block.innerHTML += "<textarea id=\"inputArea\" readonly=\"true\" rows=\"10\" cols=\"30\">  " + content + "</textarea>";

            mainBlock.appendChild(block);

        });

        function Delete(currentBlock) {

            let parent = currentBlock.parentNode;

            parent.remove();

        }

        function Save(current, object) {

            object.setAttribute("readonly", "true");

            current.remove();

        }

        function Edit(currentBlock) {

            let parent = currentBlock.parentNode;

            let currentcontent = parent.querySelector("#inputArea").value;

            parent.querySelector("#inputArea").remove();

            let textarea = document.createElement("textarea");

            textarea.id = "inputArea";

            textarea.rows = "10";

            textarea.cols = "30";

            textarea.value = currentcontent;

            let saveButton = document.createElement("button");

            saveButton.textContent = "Зберегти";

            saveButton.onclick = function () {

                Save(this, textarea);

            };

            parent.appendChild(textarea);

            parent.appendChild(saveButton);

            parent.appendChild(document.createElement("br"));

        }



Рис 6.4

**Завдання 5.**

Задати можливість заливати квадратики вибраним кольором.

Є палітра кольорів, в якій клацанням можна вибрати лише один колір. І є прямокутне полотно (контейнер, блок), яке заповнене маленькими блоками-квадратиками. При клацанні курсором на ці квадратики, вони повинні заливатись вибраним кольором.

**Лістинг коду**

**HMTL:**

    <button id="checkButton">Змінити колір</button>

    <table id="myTable">

        <tr>

            <td style="background-color: red;" id="colorRed" data-active="false" ondblclick="doubleClick(this)"></td>

            <td style="background-color: blue;" id="colorBlue" data-active="false" ondblclick="doubleClick(this)"></td>

            <td style="background-color: yellow;" id="colorYellow" data-active="false" ondblclick="doubleClick(this)">

            </td>

        </tr>

        <tr>

            <td style="background-color: green;" id="colorGreen" data-active="false" ondblclick="doubleClick(this)">

            </td>

            <td style="background-color: black;" id="colorBlack" data-active="false" ondblclick="doubleClick(this)">

            </td>

            <td style="background-color: white;" id="colorWhite" data-active="false" ondblclick="doubleClick(this)">

            </td>

        </tr>

        <tr>

            <td style="background-color: pink;" id="colorPink" data-active="false" ondblclick="doubleClick(this)"></td>

            <td style="background-color: purple;" id="colorPurple" data-active="false" ondblclick="doubleClick(this)">

            </td>

            <td style="background-color: magenta;" id="colorMagenta" data-active="false" ondblclick="doubleClick(this)">

            </td>

        </tr>

    </table>

    <table id="tablePaint"></table>

**JavaScript:**

        function doubleClick(color) {

            let activeColor = document.querySelectorAll("td");

            for (let i = 0; i < activeColor.length; i++) {

                if (activeColor[i].getAttribute("data-active") === "true") {

                    activeColor[i].setAttribute("data-active", "false");

                    activeColor[i].style.border = "4px solid white";

                }

            }

            color.style.border = "4px solid lightblue";

            color.setAttribute("data-active", "true");

        }

        let color = "";

        let checkButton = document.querySelector("#checkButton");

        color = document.querySelector("#inputColor");

        checkButton.addEventListener("click", function () {

            let allColors = document.querySelectorAll("td");

            let rectangleGradient = document.querySelector("#gradient");

            for (let i = 0; i < allColors.length; i++) {

                switch (allColors[i].id) {

                    case "colorRed": {

                        if (allColors[i].getAttribute("data-active") === "true") {

                            color = "red";

                        }

                    }

                        break;

                    case "colorBlue": {

                        if (allColors[i].getAttribute("data-active") === "true") {

                            color = "blue";

                        }

                    }

                        break;

                    case "colorYellow": {

                        if (allColors[i].getAttribute("data-active") === "true") {

                            color = "yellow";

                        }

                    }

                        break;

                    case "colorGreen": {

                        if (allColors[i].getAttribute("data-active") === "true") {

                            color = "green";

                        }

                    }

                        break;

                    case "colorBlack": {

                        if (allColors[i].getAttribute("data-active") === "true") {

                            color = "black";

                        }

                    }

                        break;

                    case "colorWhite": {

                        if (allColors[i].getAttribute("data-active") === "true") {

                            color = "white";

                        }

                    }

                        break;

                    case "colorPink": {

                        if (allColors[i].getAttribute("data-active") === "true") {

                            color = "pink";

                        }

                    }

                        break;

                    case "colorPurple": {

                        if (allColors[i].getAttribute("data-active") === "true") {

                            color = "purple";

                        }

                    }

                        break;

                    case "colorMagenta": {

                        if (allColors[i].getAttribute("data-active") === "true") {

                            color = "magenta";

                        }

                    }

                        break;

                }

            }

        });

        let tablePaint = document.querySelector("#tablePaint");

        for (let i = 0; i < 10; i++) {

            let tr = document.createElement("tr");

            for (let j = 0; j < 10; j++) {

                let td = document.createElement("td");

                td.classList.add("tdStyle");

                td.addEventListener("click", function () {

                    if (color != null) {

                        td.style.backgroundColor = color;

                    }

                });

                tr.appendChild(td);

            }

            tablePaint.appendChild(tr);

        }

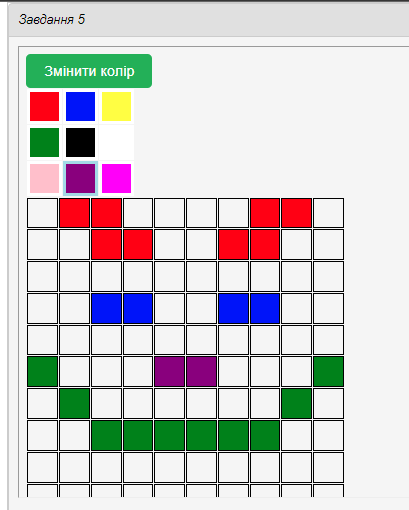


Рис 6.5

**Завдання 6.**

Є ряд гральних карт, повернутих до нас «сорочкою». При клацанні на карту, вона повертається до нас лицьовим боком протягом 1-2 секунд (повинен бути саме поворот). При повторному клацанні, карта знову повертається «сорочкою».

**Лістинг коду**

**HTML:**

<div class="card" onclick="rotateCard(this)">

        <div class="card-inner">

            <div class="card-face front"></div>

            <div class="card-face back"></div>

        </div>

    </div>

**CSS:**

       .card {

            width: 200px;

            height: 300px;

            position: relative;

            perspective: 1000px;

        }

        .card .card-inner {

            width: 200px;

            height: 300px;

            transform-style: preserve-3d;

            transition: transform 1s;

        }

        .card .card-inner .card-face {

            position: absolute;

            width: 200px;

            height: 300px;

            backface-visibility: hidden;

        }

        .card .card-inner .front {

            background-image: url("img/card\_front.png");

        }

        .card .card-inner .back {

            background-image: url("img/card\_back.png");

            transform: rotateY(180deg);

        }

        .card.face-down .card-inner {

            transform: rotateY(180deg);

        }

**JavaScript:**

function rotateCard(card) {

            card.classList.toggle('face-down');

        }



Рис 6.6

**Завдання 7.**

Створити html-сторінку зі списком повідомлень на форумі та формою для додавання нового повідомлення. Після заповнення форми додати повідомлення до списку на екрані

**Лістинг коду**

**HTML:**

   <form id="form1">

        <h1>Add comment</h1>

        <table>

            <tr>

                <td>

                    <input type="text" class="blockName" name="username">

                </td>

            </tr>

            <tr>

                <td>

                    <textarea type="text" class="blockComment" name="usercomment"></textarea>

                </td>

            </tr>

            <tr>

                <td>

                    <input type="submit" class="blockButton">

                </td>

            </tr>

        </table>

        <div id="mainBlock"></div>

    </form>

**JavaScript:**

 const formElement = document.getElementById('form1');

        let mainBlock = document.querySelector("#mainBlock");

        formElement.addEventListener('submit', (e) => {

            e.preventDefault();

            const formData = new FormData(formElement);

            const nameForm = formData.get('username');

            const commentForm = formData.get('usercomment');

            let table = document.createElement("table");

            let tr = document.createElement("tr");

            let td = document.createElement("td");

            let combinedInput = document.createElement("input");

            let nameInput = document.createElement("input");

            nameInput.type = "text";

            nameInput.value = nameForm;

            nameInput.readOnly = true;

            nameInput.classList.add("name");

            let commentTextarea = document.createElement("textarea");

            commentTextarea.value = commentForm;

            commentTextarea.readOnly = true;

            commentTextarea.classList.add("comment");

            let currentDateTime = getCurrentDateTime();

            combinedInput.value = `${currentDateTime}`;

            nameInput.style.textAlign = "left";

            combinedInput.style.textAlign = "right";

            td.appendChild(nameInput);

            td.appendChild(commentTextarea);

            td.appendChild(combinedInput);

            td.classList.add("divFix");

            tr.appendChild(td);

            table.appendChild(tr);

            mainBlock.appendChild(table);

        });

        function getCurrentDateTime() {

            const now = new Date();

            const year = now.getFullYear();

            const month = (now.getMonth() + 1).toString().padStart(2, '0');

            const day = now.getDate().toString().padStart(2, '0');

            const hours = now.getHours().toString().padStart(2, '0');

            const minutes = now.getMinutes().toString().padStart(2, '0');

            const seconds = now.getSeconds().toString().padStart(2, '0');

            return `${year}-${month}-${day} ${hours}:${minutes}:${seconds}`;

        }

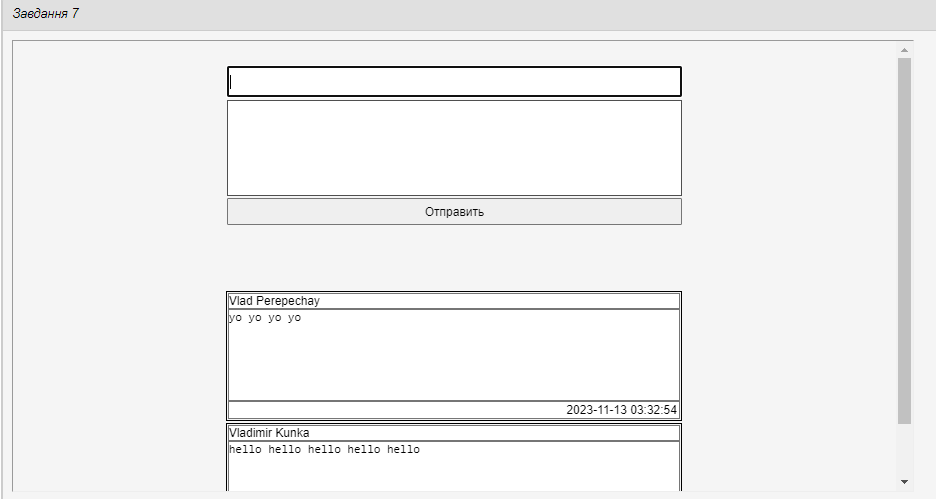


Рис 6.7

**Завдання 8.**

Створити html-сторінку зі списком книг. При натисканні на книгу, колір фону повинен змінюватися на зелений. Врахуйте, що при повторному натисканні на іншу книгу, попередньої – необхідно повертати колишній колір.

**Лістинг коду**

**HTML:**

 <div class="main">

        <ol>

            <div onclick="toggle(this)" data-active="false">

                <li>«Володар кілець» Джон Р. Р. Толкін</li>

            </div>

            <div onclick="toggle(this)" data-active="false">

                <li>«Гордість і упередження» Джейн Остін</li>

            </div>

            <div onclick="toggle(this)" data-active="false">

                <li>«Темні початки» Філіп Пулман</li>

            </div>

            <div onclick="toggle(this)" data-active="false">

                <li>«Автостопом по галактиці» Дуглас Адамс</li>

            </div>

            <div onclick="toggle(this)" data-active="false">

                <li>«Гаррі Поттер і Кубок вогню» Джоан Роулінг</li>

            </div>

            <div onclick="toggle(this)" data-active="false">

                <li>«Вбити пересмішника» Харпер Лі</li>

            </div>

        </ol>

    </div>

**JavaScript:**

function toggle(current) {

            const isActive = current.getAttribute("data-active");

            if (isActive === "false") {

                current.style.backgroundColor = "green";

                current.setAttribute("data-active", "true");

            } else {

                current.style.backgroundColor = "white";

                current.setAttribute("data-active", "false");

            }

        }

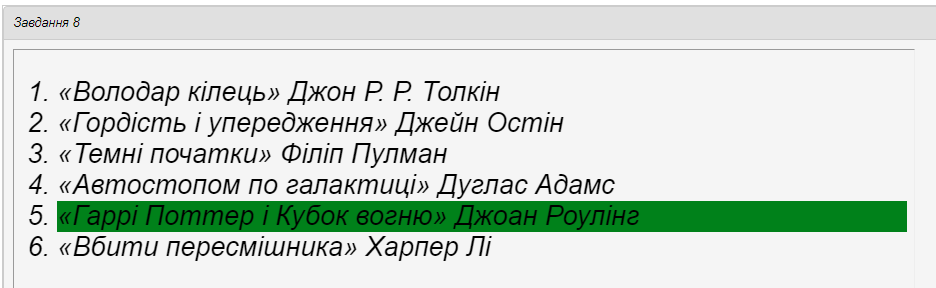


Рис 6.8