**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут **ІКНІ**

Кафедра **ПЗ**

****

**ЗВІТ**

До лабораторної роботи №1

На тему: «Структура DOM та методи доступу до вузлів дерева»

З дисципліни: «Програмування в Інтернет»

**Лектор:**

Асистент кафедри ПЗ

Степанов Д. С.

**Виконав:**

Ст. гр. ПЗ-25

Стельмах Ю. Ю.

**Прийняла:**

Старший викладач кафедри ПЗ

Грицай О. Д.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2025р.

∑=\_\_\_\_\_\_\_\_\_

Львів – 2025

**Тема:** Структура DOM та методи доступу до вузлів дерева

**Мета:** Ознайомитись з ієрархічною структурою об’єктів JavaScript та об’єктами документа і браузера

**ТЕОРЕТИЧНІ ВІДОМОСТІ**

Web-сторінка може мати вигляд дерева, вузли якого є об’єктами, до властивостей яких доступаються операторами мови програмування Javascript.

Модель DOM (Document Object Model, об’єктна модель документа) містить низку стандартних глобальних об’єктів. Зокрема, це window, navigator, document, screen, history, location.

За змістом вузли дерева поділяють на три типи:

* елементів (HTML теги) – породжені вузли або атрибути;
* тексту (текст або блоки) – не мають породжених вузлів чи атрибутів;
* атрибутів (пари атрибут/значення) – не мають породжених вузлів чи атрибутів.

За місцем розташування на дереві виділено такі типи вузлів: parent – батьківський, child – породжений, sibling – сусідній, firstchild – перший породжений, lastchild – останній породжений.

Назви вузлів є складовими назвами властивостей вузлів та параметрів методів, які ними оперують (ліквідують, модифікують, додають тощо). Вузли DOM-дерева мають такі властивості:

* firstChild, lastChild – початковий та кінцевий породжені вузли;
* childNodes – масив породжених вузлів;
* nextSibling, previousSibling – сусідні вузли (по одному), що мають одного батька;
* parentNode – елемент, що породив цей вузол.

Події – це дії, які відбуваються, внаслідок того, що робить користувач. Наприклад, якщо користувач клацає по деякій кнопці, відбувається подія Click. Якщо миша перетинає яке-небудь посилання - відбувається подія MouseOver. Існує певний набір подій, які розпізнає той чи інший браузер.

**ЗАВДАННЯ**

Сторінка має містити:

1. таблицю HTML.
2. анімацію засобами CSS3 (анімувати індикатор нових повідомлень).
3. посилання на web-сторінці.
4. зображення на web-сторінці.
5. додавання нового вузла (додавання студента по натисканню на + ).
6. видалення вузла (видалення студента кожного окремо).
7. відображення блоків при наведені на елементи (нотифікації та студент) у шапці.

**ХІД РОБОТИ**

1. Реалізація таблиці
2. <table>
3. <colgroup>
4. <col style="display: none;">
5. <col style="width: 7%">
6. <col style="width: 10%">
7. <col style="width: 20%">
8. <col style="width: 10%">
9. <col style="width: 15%">
10. <col style="width: 10%">
11. <col style="width: 15%">
12. </colgroup>
13. <tr>
14. <th class="hidden-column">Id</th>
15. <td><input id="general-checkbox" type="checkbox" aria-label="delete-checkbox"></td>
16. <th>Group</th>
17. <th>Name</th>
18. <th>Gender</th>
19. <th>Birthday</th>
20. <th>Status</th>
21. <th>Options</th>
22. </tr>
23. <tbody id="tableBody"></tbody>
24. </table>
25. <template id="rowTemplate">
26. <tr>
27. <td class="id hidden-column"></td>
28. <td><input class="delete-checkbox" type="checkbox" aria-label="delete-checkbox"></td>
29. <td class="group"></td>
30. <td class="name"></td>
31. <td class="gender"></td>
32. <td class="birthday"></td>
33. <td class="status"><img class="status-icon" src="/assets/images/offline.png" alt="offline"></td>
34. <td class="options">
35. <button class="edit-button options-button" aria-label="edit"></button>
36. <button class="delete-button options-button" aria-label="delete"></button>
37. </td>
38. </tr>
39. </template>

2. Анімація індикатора нових повідомлень

@keyframes shake {

    0% { transform: rotate(0deg); }

    25% { transform: rotate(-10deg); }

    50% { transform: rotate(10deg); }

    75% { transform: rotate(-5deg); }

    100% { transform: rotate(0deg); }

}

.notification-button:hover {

    animation: shake 0.5s ease-in-out;

}

3-4. Посилання та зображення на web-сторінці.

<nav class="sidebar">

    <div class="sidebar-buttons">

        <a class="sidebar-button" href="../dashboard.html">

            <img src="../assets/images/dashboard.png" alt="dashboard-icon">

            <span>Dashboard</span>

        </a>

        <a class="sidebar-button" href="../students.html">

            <img src="../assets/images/students.png" alt="students-icon">

            <span>Students</span>

        </a>

        <a class="sidebar-button" href="../tasks.html">

            <img src="../assets/images/clipboard.png" alt="clipboard-icon">

            <span>Tasks</span>

        </a>

    </div>

</nav>

1. Додавання нового вузла (додавання студента по натисканню на + ).

document.getElementById("add-form").addEventListener("submit", (event) => {

    event.preventDefault();

    const firstName = document.getElementById("first-name-input").value;

    const secondName = document.getElementById("last-name-input").value;

    const group = document.getElementById("group-input").value;

    const gender = document.getElementById("gender-input").value;

    const birthday = document.getElementById("birthday-input").value;

    let newStudent = {

        id: students.length + 1,

        name: firstName + " " + secondName,

        group,

        gender,

        birthday,

        status: "active"

    }

    students.push(newStudent);

    const row = template.content.cloneNode(true);

    row.querySelector(".id").textContent = newStudent.id;

    row.querySelector(".group").textContent = newStudent.group;

    row.querySelector(".name").textContent = newStudent.name;

    row.querySelector(".gender").textContent = newStudent.gender;

    row.querySelector(".birthday").textContent = formatDate(newStudent.birthday);

    tbody.appendChild(row);

    closeAddStudentModal();

    document.getElementById("add-form").reset();

    attachButtonListeners();

});

1. Видалення вузла (видалення студента кожного окремо).
2. document.querySelectorAll(".delete-button").forEach(button => {
3. button.addEventListener("click", (event) => {
4. const checkboxes = document.querySelectorAll(".delete-checkbox");
5. let checkedAtLeastOne = Array.from(checkboxes).some(checkbox => checkbox.checked);
6. const modal = document.querySelector(".delete-modal");
7. const modalText = modal.querySelector(".modal-text");
8. const okButton = modal.querySelector(".ok-button");
9. const cancelButton = modal.querySelector(".cancel-button");
10. const closeButton = modal.querySelector(".modal-close-button");
11. modal.style.display = "block";
13. if (checkedAtLeastOne) {
14. modalText.textContent = "Are you sure you want to delete these students?";
15. okButton.onclick = () => {
16. const checkedRows = [...document.querySelectorAll("#tableBody tr")].filter(row => row.querySelector(".delete-checkbox").checked);
17. checkedRows.forEach(row => deleteStudent(row));
18. modal.style.display = "none";
19. };
20. } else {
21. const row = event.target.closest("tr");
22. const studentName = row.querySelector(".name").textContent;
23. modalText.textContent = `Are you sure you want to delete ${studentName}?`;
24. okButton.onclick = () => {
25. deleteStudent(row);
26. modal.style.display = "none";
27. };
28. }
29. cancelButton.onclick = () => {
30. modal.style.display = "none";
31. };
32. closeButton.onclick = () => {
33. modal.style.display = "none";
34. }
35. });
36. });
37. function deleteStudent(row) {
38. const studentId = parseInt(row.querySelector(".id").textContent.trim());
39. const student = students.find(s => s.id === studentId);
40. if (student) {
41. students.splice(students.indexOf(student), 1);
42. row.remove();
43. }
44. }
45. Відображення блоків при наведені на елементи (нотифікації та студент) у шапці.

<header>

    <a href="../students.html" class="app-name">CMS</a>

    <div class="header-right">

        <div class="notification">

            <button id="notification-button" aria-label="notification-button" class="notification-button" onclick="window.location.href = 'messages.html'"></button>

            <div class="last-messages">

                <ul>

                    <li>

                        <img class="user-icon" src="../assets/images/user-black.png" alt="user-photo">

                        <div class="message-content">

                            <p>sender</p>

                            <p>content</p>

                        </div>

                    </li>

                    <li>

                        <img class="user-icon" src="../assets/images/user-black.png" alt="user-photo">

                        <div class="message-content">

                            <p>sender</p>

                            <p>content</p>

                        </div>

                    </li>

                    <li>

                        <img class="user-icon" src="../assets/images/user-black.png" alt="user-photo">

                        <div class="message-content">

                            <p>sender</p>

                            <p>content</p>

                        </div>

                    </li>

                </ul>

            </div>

        </div>

        <div class="user-profile">

            <button class="user-button" aria-label="user-button"></button>

            <span id="username" class="username">Yura Stelmakh</span>

            <div class="user-menu">

                <ul>

                    <li><a href="profile.html">Profile</a></li>

                    <li><a href="login.html">Log out</a></li>

                </ul>

            </div>

        </div>

    </div>

</header>

.last-messages {

    position: absolute;

    visibility: hidden;

    transform: translate(-40%, -10%);

    opacity: 0;

    transition: opacity 0.3s;

}

.notification:hover .last-messages {

    visibility: visible;

    opacity: 1;

}

.user-menu {

    position: absolute;

    transform: translate(10%, 55%);

    visibility: hidden;

    opacity: 0;

    transition: opacity 0.3s ease-in-out;

}

.user-profile:hover .user-menu {

    visibility: visible;

    opacity: 1;

}

**РЕЗУЛЬТАТ РОБОТИ**

1. Таблиця

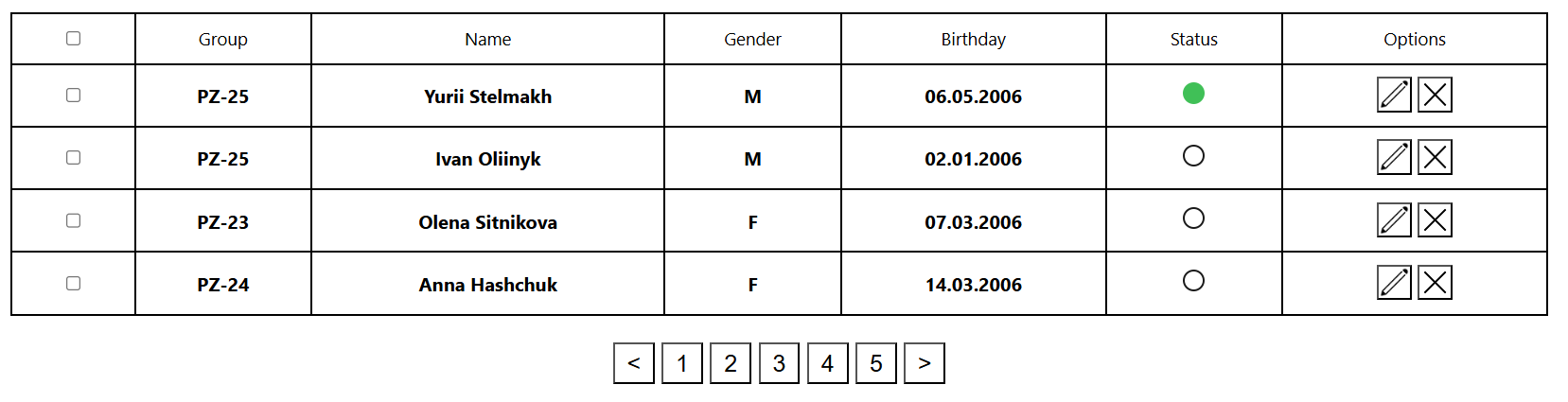


Рис. 1. Демонстрація таблиці на сторінці

1. Анімація індикатора нових повідомлень

Рис. 2. Демонстрація анімації індкатора нових повідомлень

3-4. Посилання та зображення на HTML-сторінці

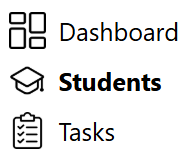


Рис. 3. Демонстрація посилань та зображень на сторінці

5. Додавання нового вузла до таблиці

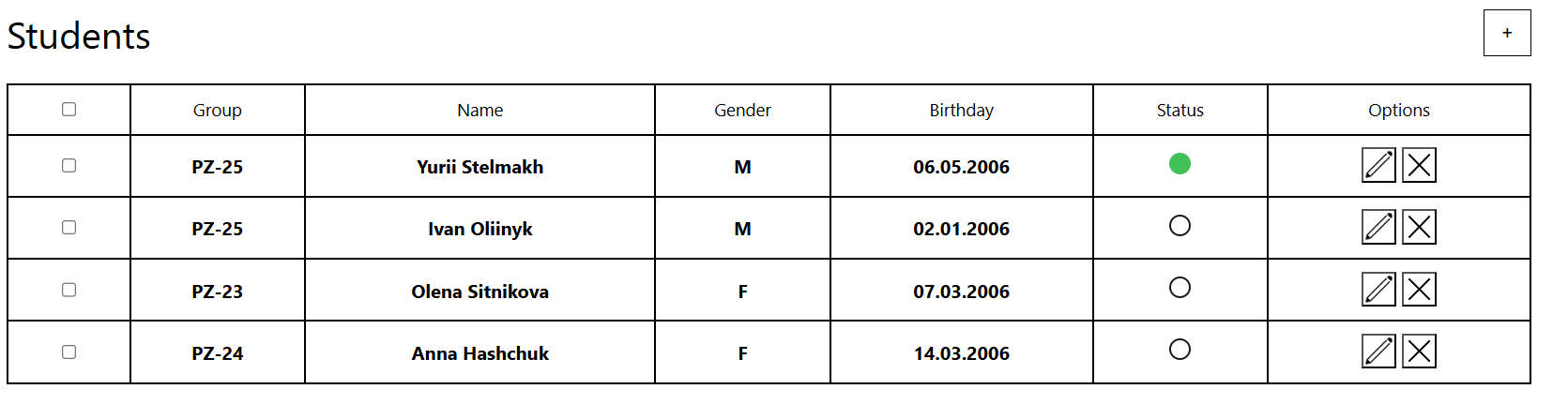


Рис. 4. Початкова таблиця

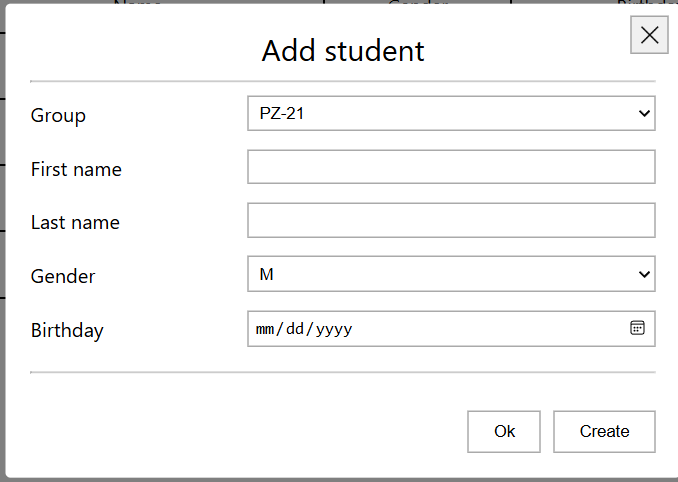


Рис. 5. Результат натиснення кнопки «+»

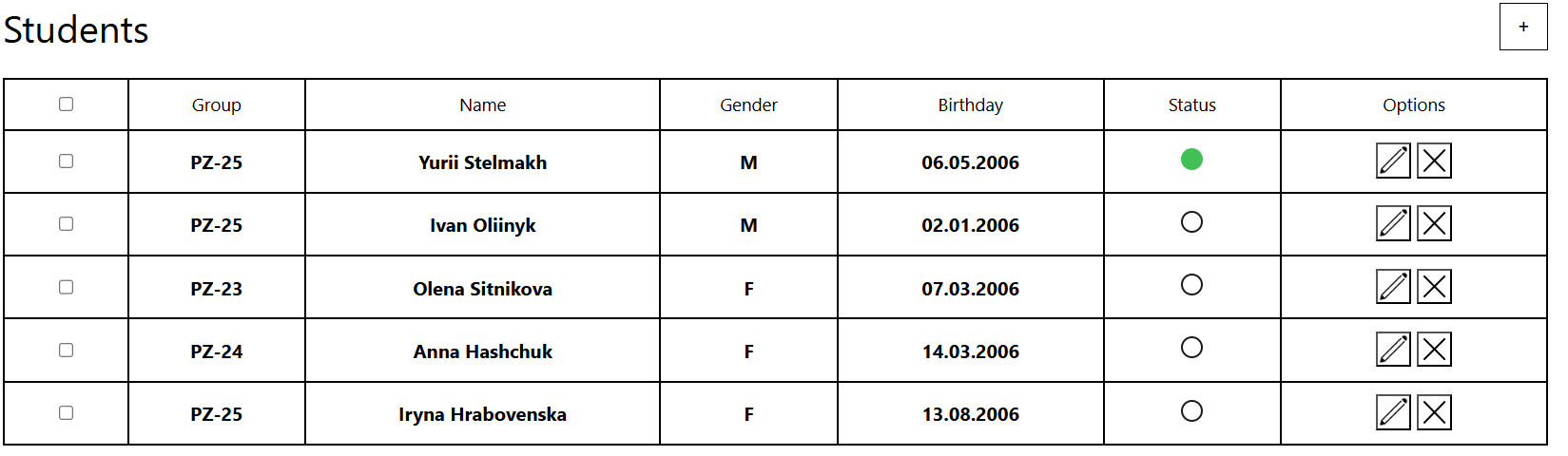


Рис. 6. Таблиця з доданим новим вузлом

6. Видалення вузла

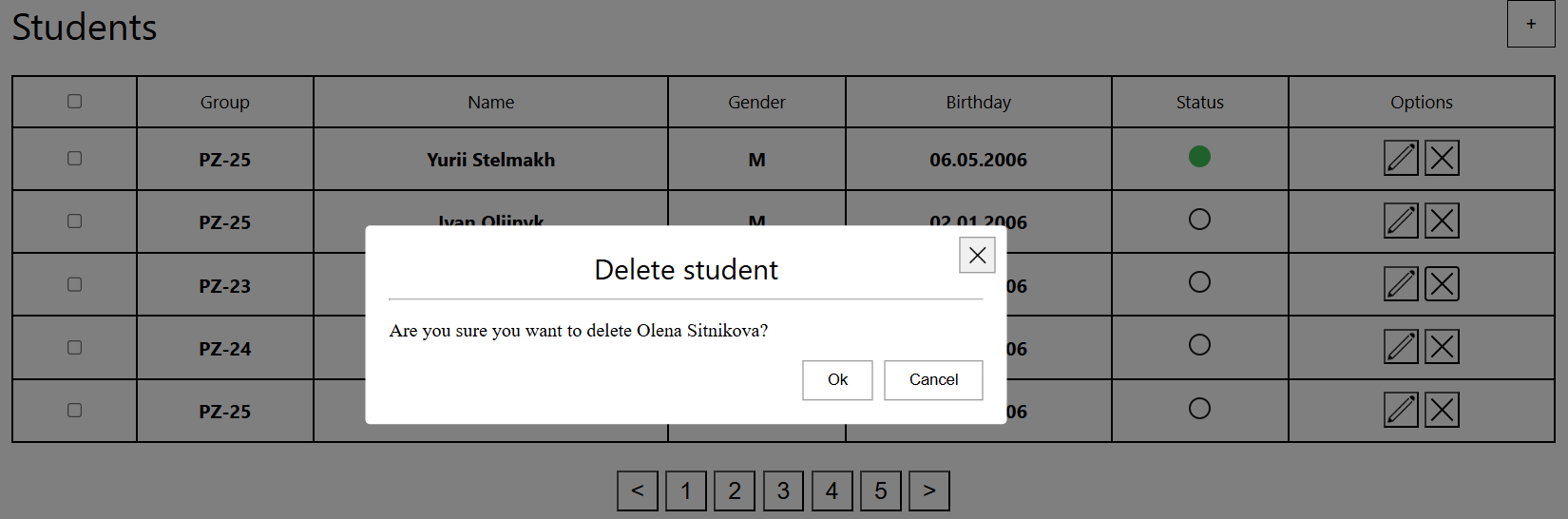


Рис. 7. Результат натиснення кнопки «x» біля елементу таблиці

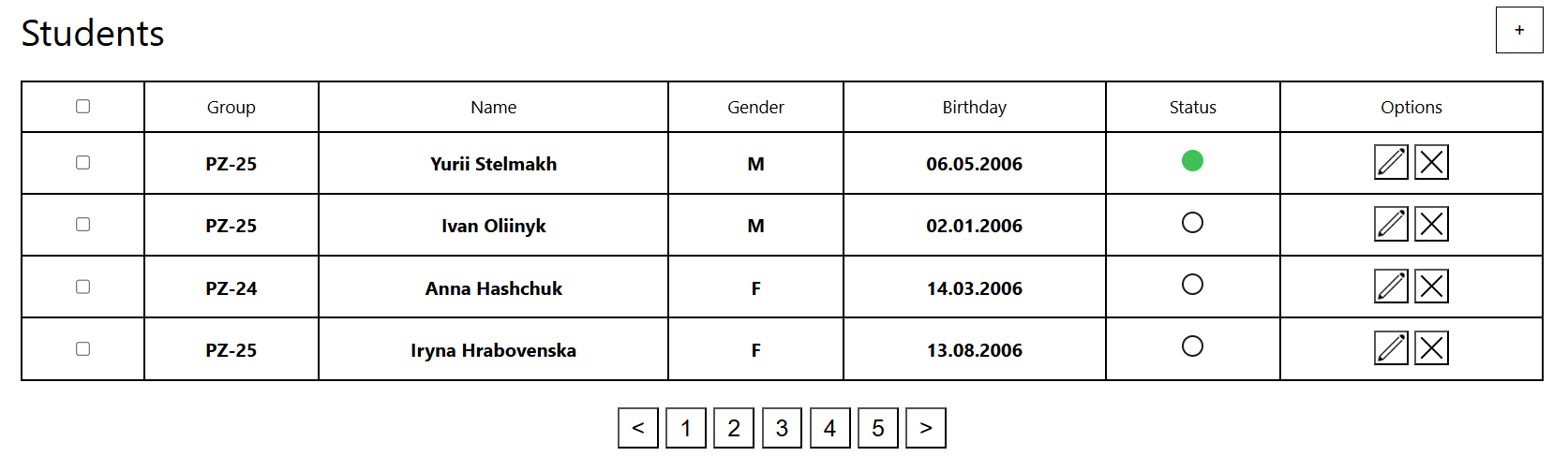


Рис. 8. Демонстрація видалення рядка

1. Відображення блоків при наведенні

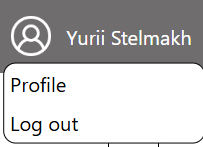
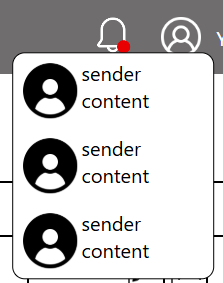


Рис. 9. Демонстрація наведення на іконку сповіщень та на іконку профіля

Посилання на проєкт: [pupsaik/PVI](https://github.com/pupsaik/PVI)

**ВИСНОВОК**

У ході виконання лабораторної роботи я створив веб-сторінку відповідно до макета, оформив її стиль за допомогою CSS та реалізував взаємодію з користувачем, додавши можливість додавання нового вузла (студента) до таблиці та його видалення. Окрім цього, я приділив увагу адаптивності, використавши media-запити.