

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

**ІКНІ  
Кафедра ПЗ**



**ЗВІТ**

До лабораторної роботи №1

**З дисципліни:** «Програмування в Інтернет»

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

**Лектор:**  
асистент каф.ПЗ  
Степанов Д.С.

**Виконав:**  
студент групи ПЗ-22  
Курман В.В.

**Прийняла:**  
старша викладачка кафедри ПЗ  
Грицай О. Д.

« \_\_\_\_ » \_\_\_\_\_ 2025 р.

Σ= \_\_\_\_ .

Львів – 2025

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

**Мета роботи:** Ознайомитись з ієрархічною структурою об'єктів JavaScript та об'єктами документа і браузера.

## Теоретичні відомості

### Модель DOM

Web-сторінка може мати вигляд дерева, вузли якого є об'єктами, до властивостей яких доступуються 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. Ієрархію основних об'єктів у моделі DOM подано на рис. 1.2.

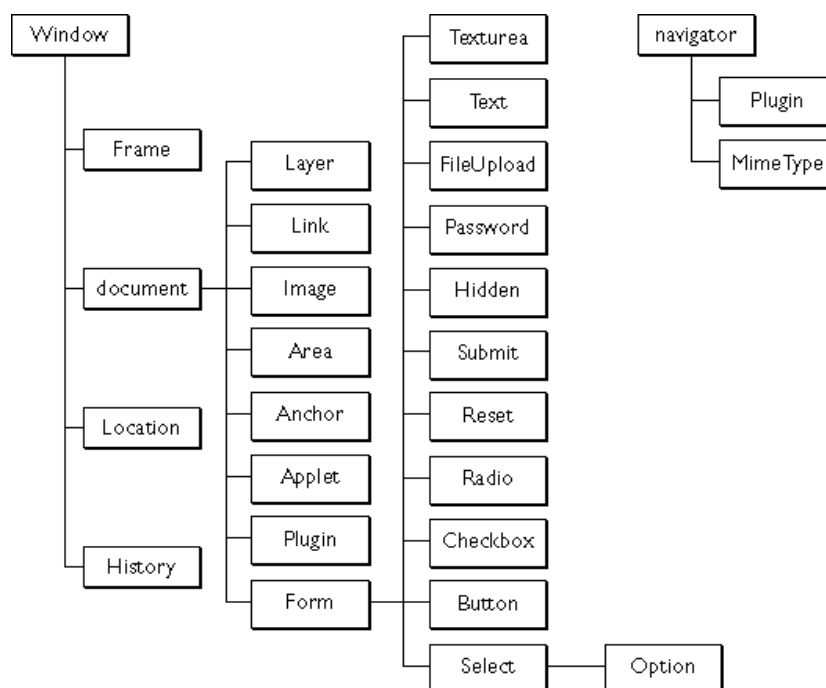


Рис. 1.2. Дерево моделі DOM об'єкта window браузера

Перерахуємо глобальні об'єкти та їх призначення, а також розглянемо методи та властивості глобальних об'єктів DOM:

- window – вікно браузера, найвищий об'єкт ієрархії DOM. Усі глобальні змінні стають частинами об'єкта window. Його методи такі:

alert, blur, clearInterval, clearTimeout, close, confirm, focus, moveBy, moveTo, open, print, prompt, resizeBy, resizeTo, scrollBy, scrollTo, setInterval, setTimeout

Властивості: document, history, location, name

- navigator – інформація про браузер.

Властивості: appName, appVersion, browserLanguage, cookieEnabled, platform, userAgent

- screen – інформація про екран, який використовує браузер.

Властивості: availHeight, availWidth, colorDepth, height, pixelDepth, width

- history – список викликаних сторінок.

Властивості: length

Методи: back, forward, go

- location – URL поточної web-сторінки.

Властивості: host, hostname, href, pathname, port, protocol, search

Методи: assign, reload, replace

- document – поточна Web-сторінка.

Властивості: anchors, body, cookie, domain, forms, images, links, referrer, title, URL

Методи: close, getElementById, getElementsByName, open, write, writeln, getElementsByTagName

Приклад ієрархічного дерева DOM для об'єкта `document` наведено на рис. 1.3.

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.

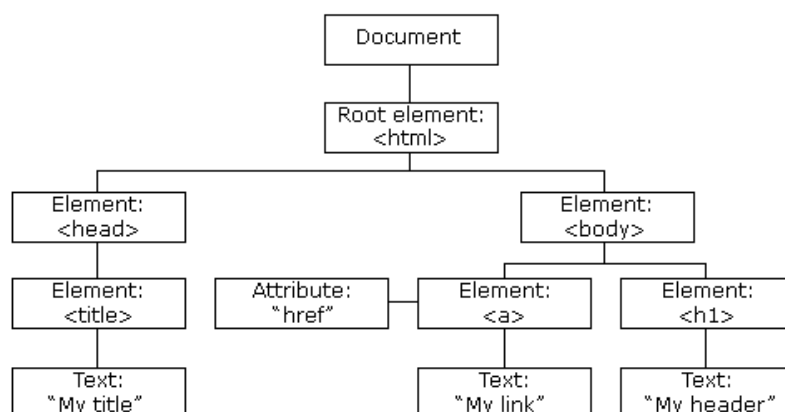
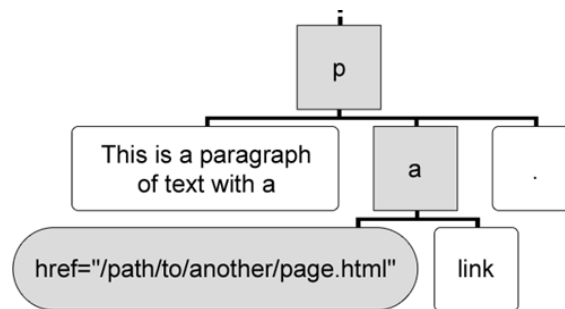


Рис. 1.3. Дерево об'єкта `document` у моделі DOM

Наведемо фрагмент сторінки і відповідне йому дерево (рис. 1.4)

`<p>This is a paragraph of text with a  
<a href="/path/to/another/page.html">link inside </a>.`



*Рис. 1.4. Фрагмент дерева з вузлом параграфа*

### Типи вузлів та методи доступу до них

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.

Мова програмування дає змогу змінювати склад вузлів, вигляд та зміст web-сторінки. На моделі DOM програміст має можливість виконувати операції двох типів: на рівні вузлів для зміни структури дерева додаванням або відніманням вузла та заміни вузла; на рівні вузла для зміни змісту елемента. Зокрема, для першого типу операцій застосовуються методи:

- appendChild(node): розміщає вузол у кінці списку породжених вузлів до заданого;
- insertBefore(newChild, oldChild): розміщає новий вузол перед вузлом oldChild у списку породжених вузлів до заданого;
- removeChild(node): знищує наведений вузол зі списку породжених вузлів до заданого;
- replaceChild(newChild, oldChild): заміняє породжений на новий вузол.

Властивості набувають значення

- `nodeType`: 1 – елемент, 2 – атрибут, 3 – текст, 4 – коментарі, 9 – документ.

- `nodeName`: повертає версію тега, наприклад, "div" чи "article". Текстові вузли мають назву "#text". Назва вузла document є "#document",

- `nodeValue`: текст вузла або значення атрибута.

Доступ до вузлів DOM за тегами чи ідентифікаторами здійснюється методами:

- `document.getElementById("id")` – елемент;

- `element.getElementsByTagName("tag")` – усі породжені вузли;

`document.createElement("tag")` - команда створює новий порожній вузол для подання елемента певного типу. Рис.1. Загальна схема ієрархічної структури об'єктів.

### Опрацювання подій

Події та обробники подій є дуже важливою частиною у програмування на JavaScript. **Події** (*Events*), головним чином, ініціюються тими або іншими діями користувача. 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.

Ми можемо примусити нашу JavaScript-програму реагувати на деякі з них. І це може бути виконано за допомогою спеціальних програм **обробки подій**. Так, в результаті клацання по кнопці може створюватися випадające вікно. Це означає, що створення вікна повинно бути реакцією на подію Click. Програма - обробник подій, яку ми повинні використати в даному випадку, називається `onClick`. І вона повідомляє комп'ютер, що потрібно робити, якщо відбудеться дана подія.

Обробник подій записується в документ як атрибут тега HTML, до якого ви приписуєте код JavaScript. Наприклад, ви створили **функцію** JavaScript, і назвали її `compute`. Ви можете примусити браузер виконувати цю функцію, коли користувач натискає на кнопку, в якій до `onClick` приписаний обробник результату кнопки:

```
INPUT TYPE="button" VALUE="Calculate"
onClick="compute(this.form)"
```

Ви можете поміщати будь-які твердження JavaScript усередині кавичок `onClick`. Ці твердження будуть виконані, коли користувач натискатиме на

кнопку. Якщо Ви хочете включити більш ніж одне твердження, то окремі твердження записуються через крапку з комою (;).

Взагалі, це – непогана ідея визначати *функцію* для обробників подій тому що:

- це робить ваш код мобільним, оскільки ви можете використовувати ту ж саму функцію в багатьох різних місцях.
- це робить ваші твердження більш легкими для читання.

В даному прикладі використовується `this.form`, щоб звернутися до поточної форми. Ключове слово звертається до об'єкту (об'єкту кнопки у вищезазначеному прикладі). Потім конструкція `this.form` звертається до форми, що містить кнопку. Далі – обробник події `onClick` робить запит до функції `compute()`, з поточною формою `this.form`, як параметр функції.

Події звертаються до тегів HTML таким чином:

- події **Focus, Blur, Change**: text fields, textareas, і selections;
- подія **Click**: buttons, radio buttons, checkboxes, submit buttons, reset buttons, links;
- подія **Select**: text fields, textareas;
- подія **MouseOver**: links.

Ви можете використовувати в скрипті безліч різних типів функцій обробки подій. Щоб дізнатись про усі існуючі обробники подій, звертайтеся до відповідного довідника. Наведемо лише деякі з них:

- **onLoad** - виконання скрипта або функції при завантаженні;
- **onChange** - породжується при зміні значення елемента форми;
- **onClick** - породжується при виборі об'єкту (button, checkbox і т.п.);
- **onSelect** - породжується при виборі текстового об'єкту (text, textarea);
- **onSubmit** - при натисненні на кнопку Submit;
- **onUnload** - при переході до іншої сторінки.

### Завдання

1. Розробити web-сторінку згідно макета (wireframe).
2. Сторінка повинна відповідати наступним вимогам.

### Виконання

## Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

### Showing results for uploaded file index.html

Checker Input

Show ☐ source ☐ outline ☐ image report

Check by

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 10 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 25.4.18


## Рис. 1. Результат валідації сайту на <https://validator.w3.org/>

W3C результати перевірки CSS для main.css (CSS3 + SVG)

**Вітасмо! Помилки не виявлено.**


Цей документ проходить перевірку за стандартом **CSS3 + SVG** !

Для того, щоб продемонструвати відвідувачам, що дбаєте про створення коректних сторінок, ви можете розмістити цей знак на будь-якій перевірній сторінці. Ось код XHTML для розміщення:



```

```



```

```


(закрийте тег img, використовуючи > замість />, якщо використовуєте HTML <= 4.01)

W3C результати перевірки CSS для style.css (CSS3 + SVG)

**Вітасмо! Помилки не виявлено.**


Цей документ проходить перевірку за стандартом **CSS3 + SVG** !

Для того, щоб продемонструвати відвідувачам, що дбаєте про створення коректних сторінок, ви можете розмістити цей знак на будь-якій перевірній сторінці. Ось код XHTML для розміщення:



```

```



```

```

(закрийте тег img, використовуючи > замість />, якщо використовуєте HTML <= 4.01)

## Рис. 2-3. Результат валідації сайту на <https://jigsaw.w3.org/css-validator/>

# CSS LINT

Will hurt your feelings\*  
(And help you code better)

CSS lint found **0** errors and **19** warnings. How can you fix it? See the details below.

RESTART

# CSS LINT

Will hurt your feelings\*  
(And help you code better)

CSS lint found **0** errors and **28** warnings. How can you fix it? See the details below.

RESTART

Рис. 4-5. Результат валідації сайту на <https://csslint.net/>

## CONFIGURE

### Metrics

There are **17** functions in this file.

Function with the largest signature take **1** arguments, while the median is **0**.

Largest function has **19** statements in it, while the median is **3**.

The most complex function has a cyclomatic complexity value of **7** while the median is **1**.



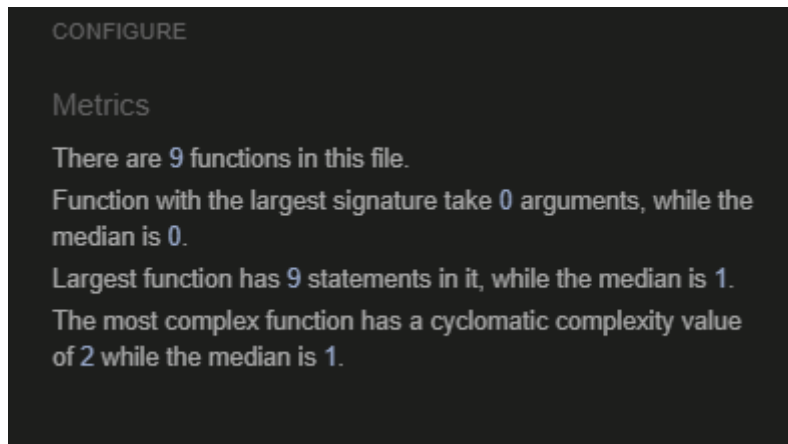


Рис. 6-7. Результат валідації сайту на <https://jshint.com/>

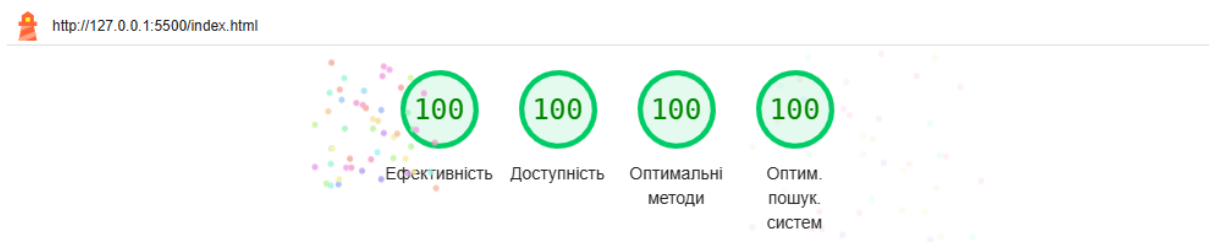


Рис. 8. Результат валідації сайту на Lighthouse

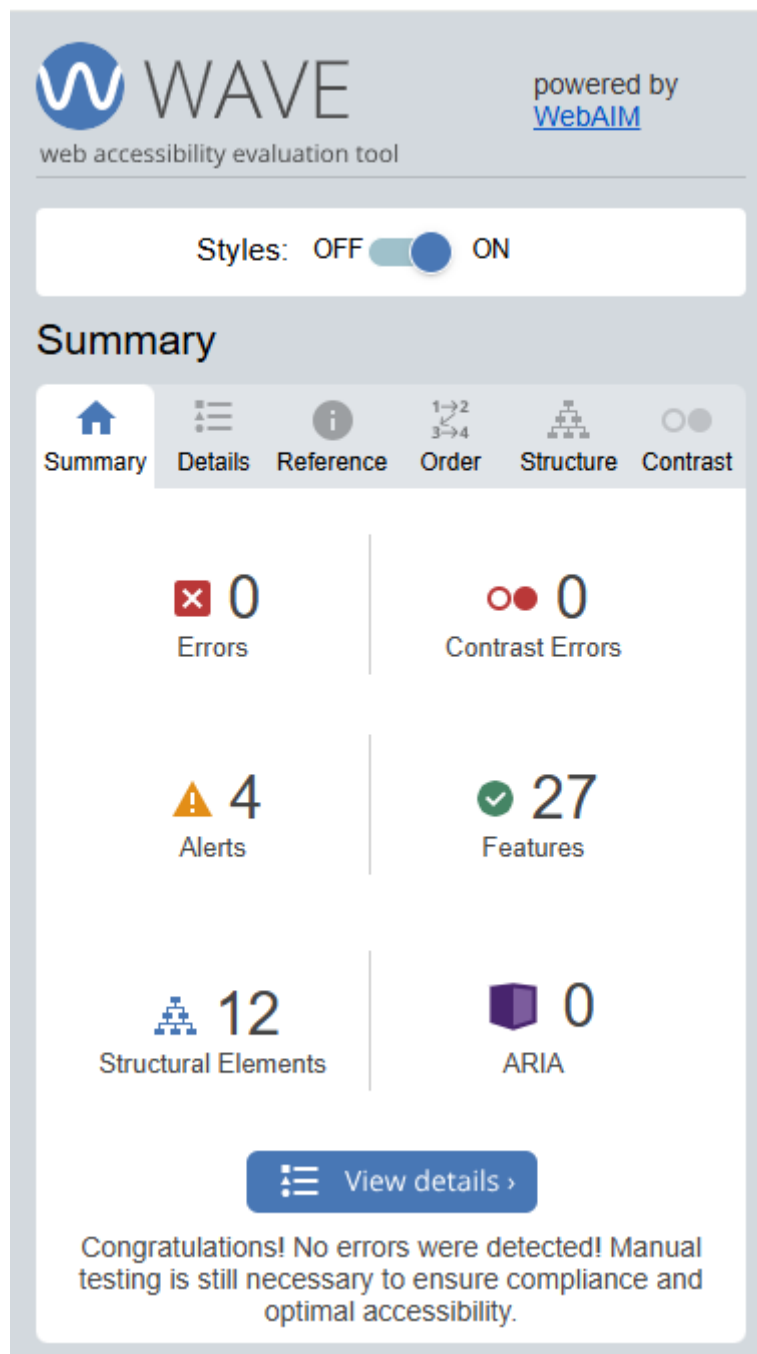


Рис. 9. Результат валідації сайту на <https://wave.webaim.org/>











CMS

Dashboard

Students

Tasks

## Students

| <input type="checkbox"/> | Group | Name          | Gender | Birthday   | Status | Options   |
|--------------------------|-------|---------------|--------|------------|--------|---|
| <input type="checkbox"/> | KN-21 | John Smith    | M      | 11.05.2004 | ●      |   |
| <input type="checkbox"/> | KN-22 | Ann Clerk     | F      | 11.03.2001 | ●      |   |
| <input type="checkbox"/> | KN-22 | Alex Johnson  | M      | 05.12.1995 | ●      |   |
| <input type="checkbox"/> | KN-22 | Emily Davis   | F      | 22.09.2000 | ●      |   |
| <input type="checkbox"/> | KN-22 | Michael Brown | M      | 15.07.1998 | ●      |   |

<

1

2

3

4

>











CMS

Dashboard

Students

Tasks

## Students

| <input type="checkbox"/> | Group |  | Birthday   | Status | Options   |
|--------------------------|-------|--|------------|--------|---|
| <input type="checkbox"/> | KN-21 |  | 11.05.2004 | ●      |     |
| <input type="checkbox"/> | KN-22 |  | 11.03.2001 | ●      |     |
| <input type="checkbox"/> | KN-22 |  | 05.12.1995 | ●      |     |
| <input type="checkbox"/> | KN-22 |  | 22.09.2000 | ●      |     |
| <input type="checkbox"/> | KN-22 |  | 15.07.1998 | ●      |   |

×

Add student

Group

Select Group

First name

Last name

Gender

Select Gender

Birthday

dd . mm . yyyy

Cancel

Create











CMS

Dashboard

Students

Tasks

## Students

| <input type="checkbox"/> | Group |  | Birthday   | Status | Options   |
|--------------------------|-------|--|------------|--------|---|
| <input type="checkbox"/> | KN-21 |  | 11.05.2004 | ●      |   |
| <input type="checkbox"/> | KN-22 |  | 11.03.2001 | ●      |   |
| <input type="checkbox"/> | KN-22 |  | 05.12.1995 | ●      |   |
| <input type="checkbox"/> | KN-22 |  | 22.09.2000 | ●      |   |
| <input type="checkbox"/> | KN-22 |  | 15.07.1998 | ●      |   |

×

Add student

Group

KN-21

First name

Oleg

Last name

Smiths

Gender

Male

Birthday

09 . 04 . 2025

Cancel

Create

CMS

Dashboard

Students

Tasks

Students

+

| <input type="checkbox"/> | Group | Name          | Gender | Birthday   | Status      | Options                           |
|--------------------------|-------|---------------|--------|------------|-------------|-----------------------------------|
| <input type="checkbox"/> | KN-21 | John Smith    | M      | 11.05.2004 | <div></div> | <div><div></div><div></div></div> |
| <input type="checkbox"/> | KN-22 | Ann Clerk     | F      | 11.03.2001 | <div></div> | <div><div></div><div></div></div> |
| <input type="checkbox"/> | KN-22 | Alex Johnson  | M      | 05.12.1995 | <div></div> | <div><div></div><div></div></div> |
| <input type="checkbox"/> | KN-22 | Emily Davis   | F      | 22.09.2000 | <div></div> | <div><div></div><div></div></div> |
| <input type="checkbox"/> | KN-22 | Michael Brown | M      | 15.07.1998 | <div></div> | <div><div></div><div></div></div> |
| <input type="checkbox"/> | KN-21 | Oleg Smiths   | M      | 09.04.2025 | <div></div> | <div><div></div><div></div></div> |

<

1

2

3

4

>

CMS

Dashboard

Students

Tasks

Students

+

| <input type="checkbox"/>            | Group | Name         | Gender | Birthday   | Status      | Options                           |
|-------------------------------------|-------|--------------|--------|------------|-------------|-----------------------------------|
| <input type="checkbox"/>            | KN-21 | John Smith   | M      | 11.05.2004 | <div></div> | <div><div></div><div></div></div> |
| <input type="checkbox"/>            | KN-22 | Ann Clerk    | F      | 11.03.2001 | <div></div> | <div><div></div><div></div></div> |
| <input checked="" type="checkbox"/> | KN-22 | Alex Johnson | M      | 05.12.1995 | <div></div> | <div><div></div><div></div></div> |
| <input type="checkbox"/>            | KN-22 | Emily Davis  | F      | 22.09.2000 | <div></div> | <div><div></div><div></div></div> |
| <input type="checkbox"/>            | KN-22 |              |        | 15.07.1998 | <div></div> | <div><div></div><div></div></div> |
| <input type="checkbox"/>            | KN-21 |              |        | 09.04.2025 | <div></div> | <div><div></div><div></div></div> |

Warning

Are you sure you want to delete user "Alex Johnson"?

Cancel

OK

CMS

Dashboard

Students

Tasks

Students

+

| <input type="checkbox"/>            | Group | Name          | Gender | Birthday   | Status      | Options                           |
|-------------------------------------|-------|---------------|--------|------------|-------------|-----------------------------------|
| <input type="checkbox"/>            | KN-21 | John Smith    | M      | 11.05.2004 | <div></div> | <div><div></div><div></div></div> |
| <input type="checkbox"/>            | KN-22 | Ann Clerk     | F      | 11.03.2001 | <div></div> | <div><div></div><div></div></div> |
| <input checked="" type="checkbox"/> | KN-22 | Alex Johnson  | M      | 05.12.1995 | <div></div> | <div><div></div><div></div></div> |
| <input type="checkbox"/>            | KN-22 | Emily Davis   | F      | 22.09.2000 | <div></div> | <div><div></div><div></div></div> |
| <input type="checkbox"/>            | KN-22 | Michael Brown | M      | 15.07.1998 | <div></div> | <div><div></div><div></div></div> |
| <input type="checkbox"/>            | KN-21 | Oleg Smiths   | M      | 09.04.2025 | <div></div> | <div><div></div><div></div></div> |

<

1

2

3

4

>

Admin

John K.

Ann S.

Via live on the most beautiful planet, Earth which has very clean and attractive nature full of greenery.

Nature is our best friend which provides us all the resources to live here.

If gives us water to drink, pure air to breathe, food to eat, land to stay, animals, plants for our other uses, etc for our betterment.

CMS

Dashboard

Students













Tasks

Students

James Bond

Profile

Log Out

| <input type="checkbox"/>            | Group | Name          | Gender | Birthday   | Status | Options   |
|-------------------------------------|-------|---------------|--------|------------|--------|---|
| <input type="checkbox"/>            | KN-21 | John Smith    | M      | 11.05.2004 | ●      |   |
| <input type="checkbox"/>            | KN-22 | Ann Clerk     | F      | 11.03.2001 | ●      |   |
| <input checked="" type="checkbox"/> | KN-22 | Alex Johnson  | M      | 05.12.1995 | ●      |   |
| <input type="checkbox"/>            | KN-22 | Emily Davis   | F      | 22.09.2000 | ●      |   |
| <input type="checkbox"/>            | KN-22 | Michael Brown | M      | 15.07.1998 | ●      |   |
| <input type="checkbox"/>            | KN-21 | Oleg Smiths   | M      | 09.04.2025 | ●      |   |

<

1

2

3

4

>

CMS

Dashboard

Students

Tasks













Students

James Bond

Profile

Log Out

+

| <input type="checkbox"/>            | Group | Name          | Gender | Birthday   | Status | Options   |
|-------------------------------------|-------|---------------|--------|------------|--------|---|
| <input type="checkbox"/>            | KN-21 | John Smith    | M      | 11.05.2004 | ●      |   |
| <input type="checkbox"/>            | KN-22 | Ann Clerk     | F      | 11.03.2001 | ●      |   |
| <input checked="" type="checkbox"/> | KN-22 | Alex Johnson  | M      | 05.12.1995 | ●      |   |
| <input type="checkbox"/>            | KN-22 | Emily Davis   | F      | 22.09.2000 | ●      |   |
| <input type="checkbox"/>            | KN-22 | Michael Brown | M      | 15.07.1998 | ●      |   |
| <input type="checkbox"/>            | KN-21 | Oleg Smiths   | M      | 09.04.2025 | ●      |   |

<

1

2

3

4

>

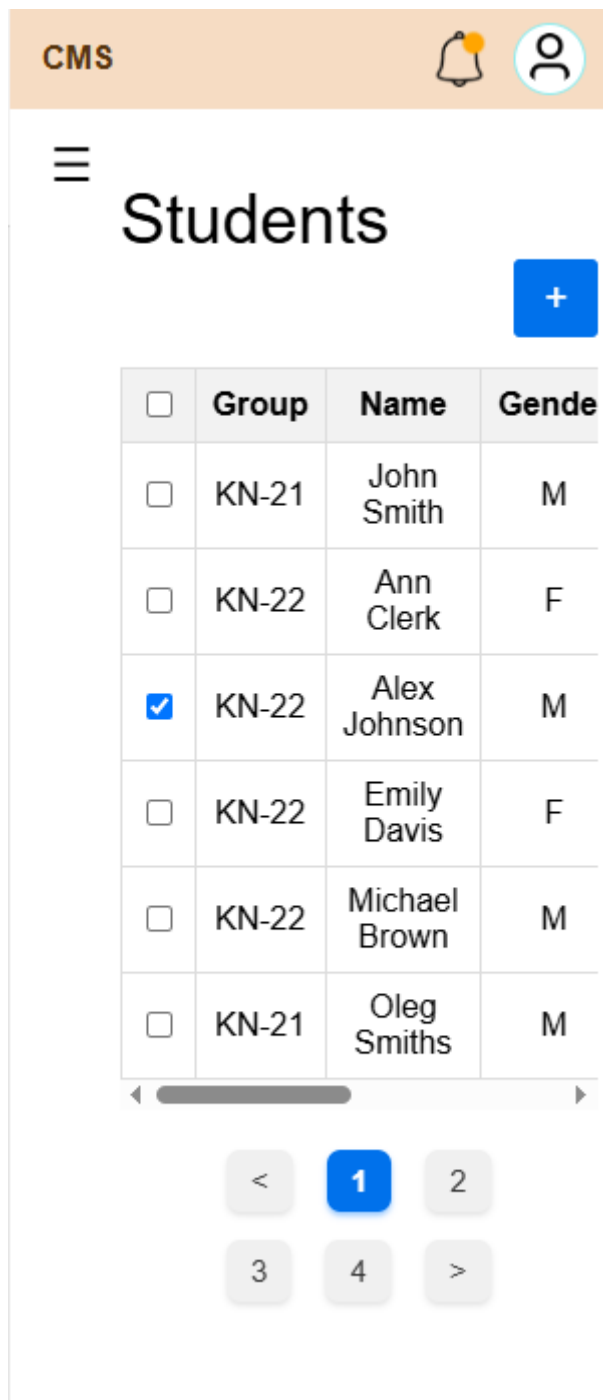


Рис. 10-18. Сайт

### Висновки

На даній лабораторній роботі, я ознайомився з ієрархічною структурою об'єктів JavaScript та об'єктами документа і браузера та методами доступу до вузлів дерева. В ході виконання лабораторної роботи Було розроблено інтерактивний інтерфейс, який забезпечує динамічну зміну контенту за допомогою скриптів, а

також реалізовано адаптивне відображення елементів для різних розмірів екранів. Завдяки цьому проєкту вдалося краще зрозуміти принципи структурованої розмітки, стилізації через каскадні таблиці стилів та програмної взаємодії за допомогою JavaScript, що є важливою основою для подальшого розвитку у сфері веб-розробки.