

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи №6 з дисципліни
«Програмування веб-застосунків»

«Взаємодія фронтенду та бекенду»

Варіант 13

Виконав

ІП-33 Хребтань Максим Віталійович

(шифр, прізвище, ім'я, по батькові)

Перевірив

Очеретяний Олександр Костянтинович

(прізвище, ім'я, по батькові)

Київ 2024

ЗМІСТ

ЛАБОРАТОРНА РОБОТА №6	3
Взаємодія фронтенду та бекенду	3
ХІД РОБОТИ	5
Посилання	14
ВИСНОВОК	15

ЛАБОРАТОРНА РОБОТА №6

Взаємодія фронтенду та бекенду

Мета роботи: Організація бекенду і фронтенду за допомогою Javascript та PHP.

Завдання:

1. Оберіть сервер (хостінг або сервіс розгортання веб-додатків) для розміщення функціонального прикладу виконаного завдання.
2. За допомогою HTML, CSS та мов програмування Javascript і PHP виконайте маніпуляції з веб-сторінками згідно з вашим варіантом:
 - a. Створіть дві копії веб-сторінки із 3 л.р. – це будуть перша та друга веб-сторінки в даній л.р.
 - b. В центральному блоці («3», «4» або «5») першої веб-сторінки за допомогою Javascript та елементів форми організуйте створення об'єкта (набору об'єктів) згідно із варіантом завдання. Кількість елементів, порядок їхнього розташування і контент в них задає користувач першої сторінки.
 - c. За допомогою Javascript та PHP організуйте збереження в асинхронному режимі даних об'єкта чи набору об'єктів (технічних параметрів та контенту) на сервері у вибраному вами форматі (файли, БД тощо).
 - d. В центральному блоці («3» , «4» або «5») другої веб-сторінки за допомогою Javascript та PHP забезпечте отримання даних об'єкта (набору об'єктів) із сервера і формування коду на HTML, CSS та JS, необхідного для функціонування об'єкта (набору об'єктів). Сформований об'єкт (набір об'єктів) повинен функціонувати без використання JS та CSS-фреймворків (jQuery, Bootstrap та ін.).
 - e. Організуйте періодичний асинхронний контроль змін в даних об'єкта (набору об'єктів) на сервері для регулярного оновлення об'єкта (набору об'єктів) на другій сторінці.
3. Для запису стилів і скриптів використовуйте будь-які відомі вам способи.
4. Файли виконаного завдання розмістіть в репозиторії на Github, функціональний приклад виконаного завдання розмістіть на сервері та продемонструйте його роботу або зафіксуйте його роботу на відео.

Варіант 13

Набір об'єктів «Dropdown», приклад меню сайта:

<https://get.foundation/index.html>.

Змінювані параметри – елементи горизонтального меню, випадаючі вниз пункти меню, посилання на сторінки для пунктів меню.

ХІД РОБОТИ

Перша сторінка:

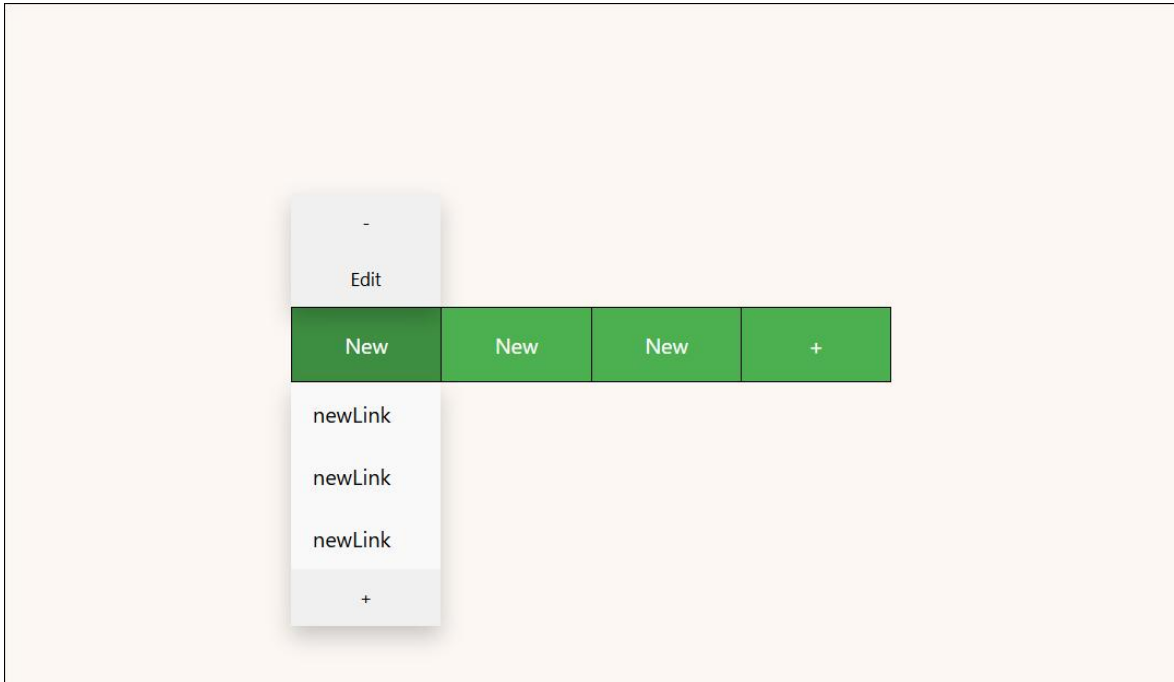


Рис. 1. Створений елемент

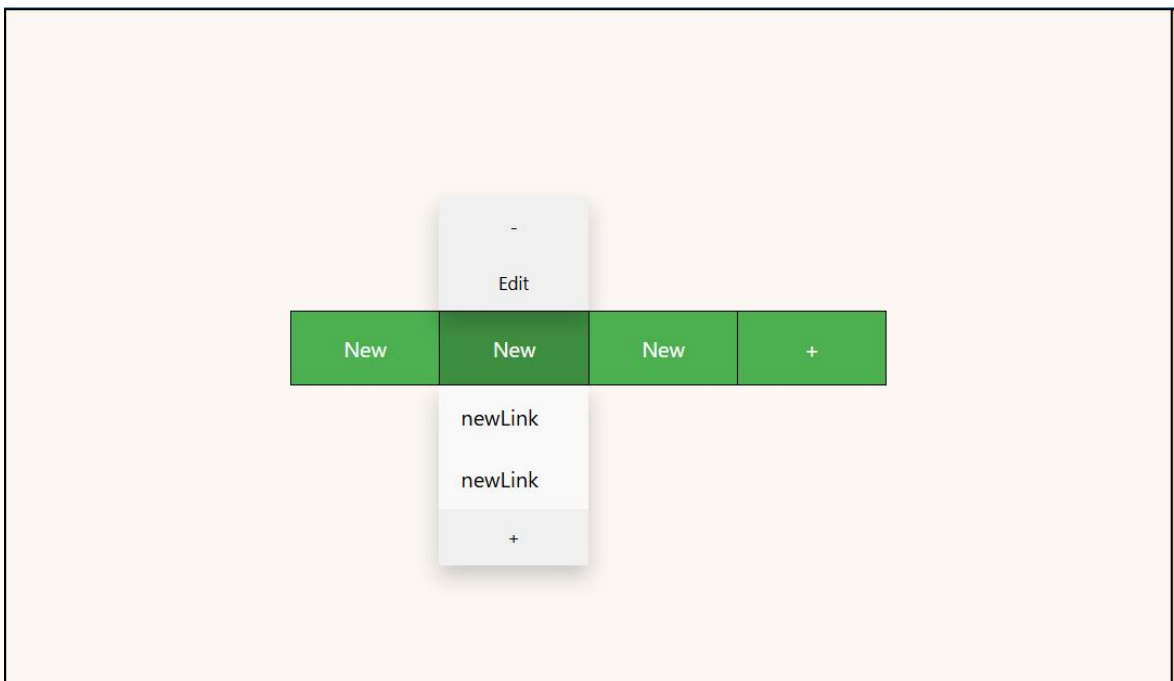


Рис. 2. Створений елемент



Рис. 3. Створений елемент

Друга сторінка:

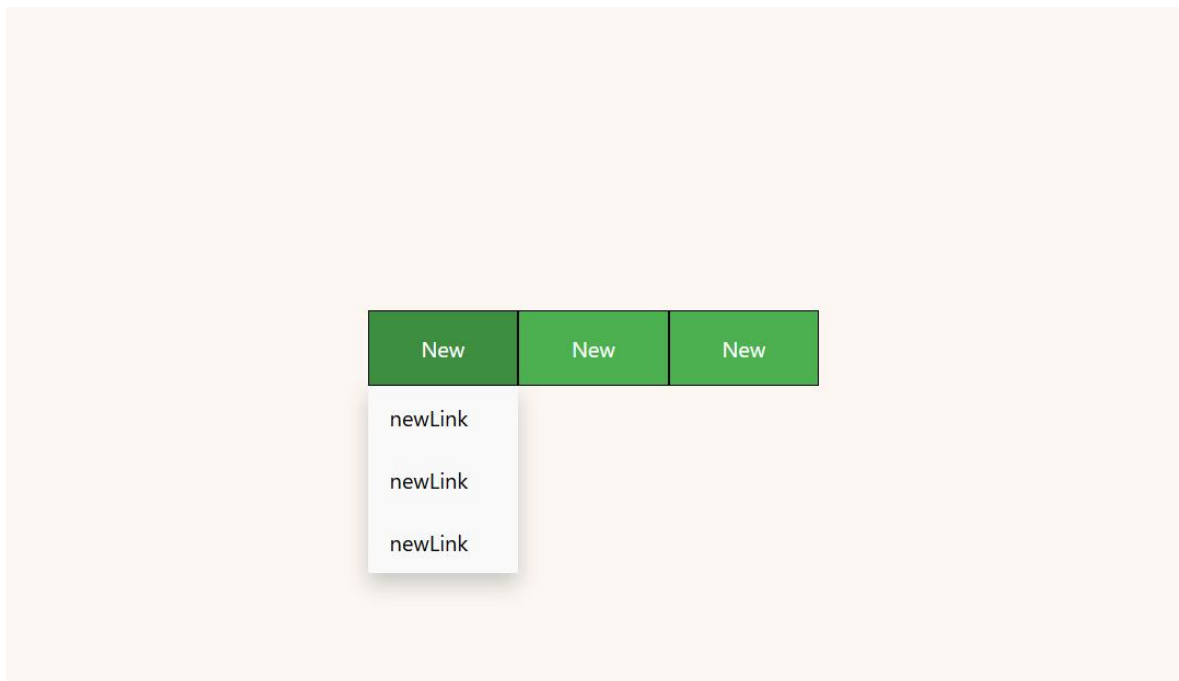


Рис. 4. Отриманий елемент з серверу

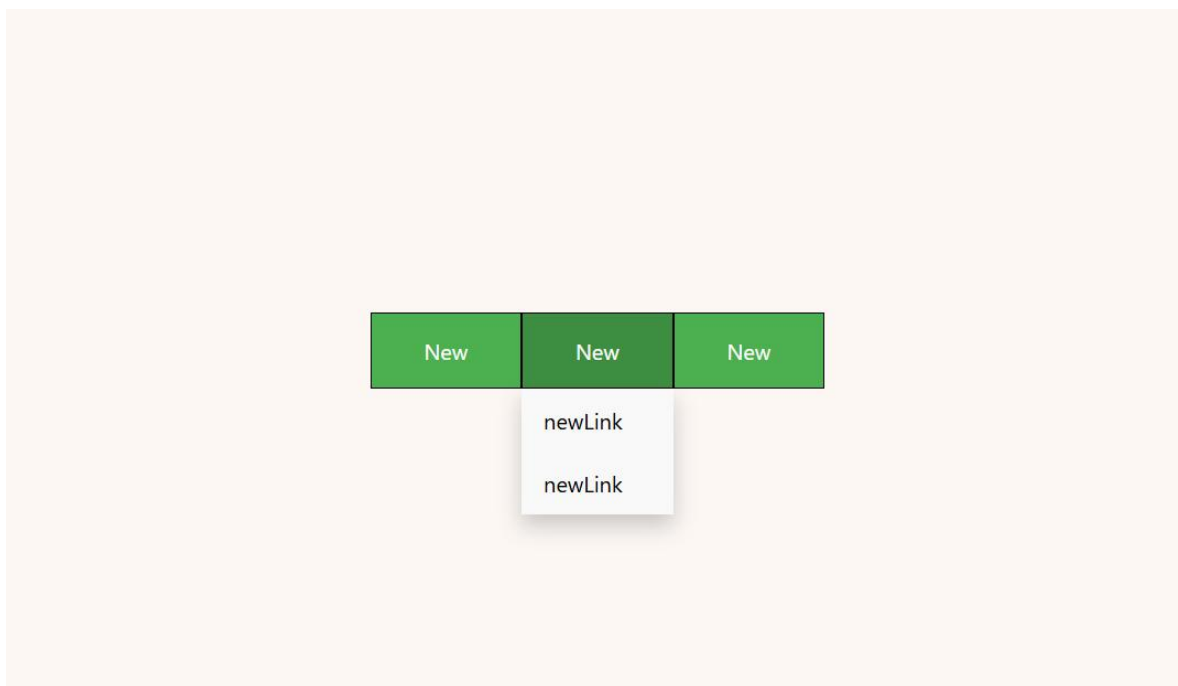


Рис. 5. Отриманий елемент з серверу

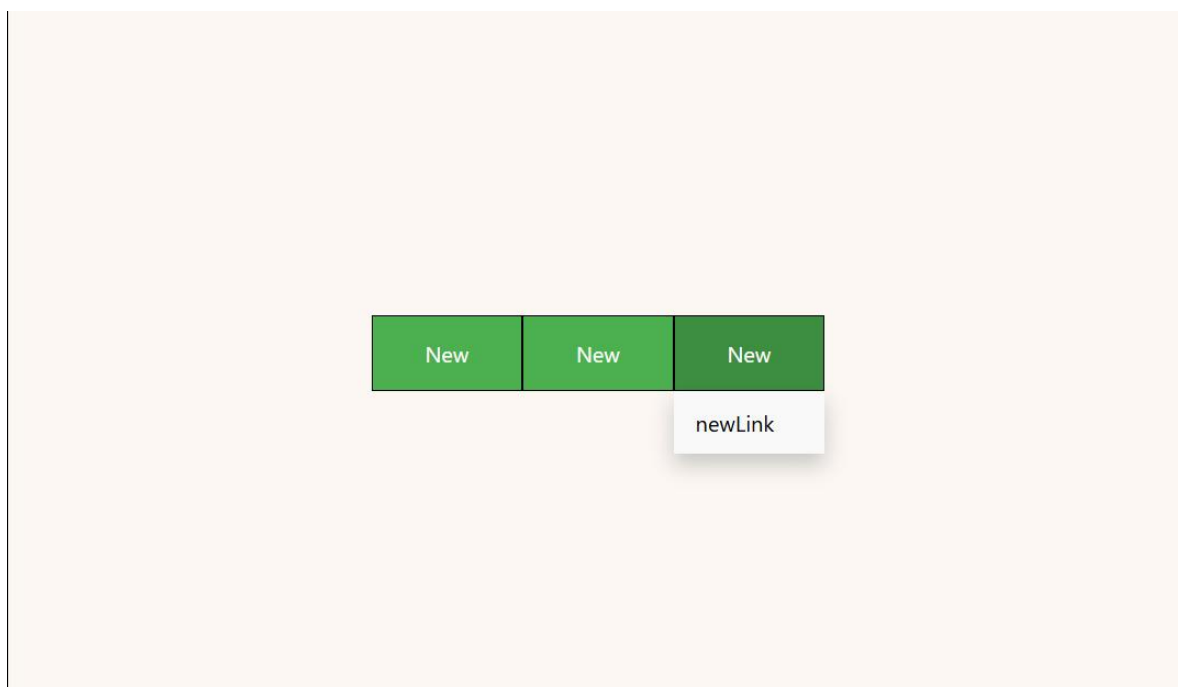


Рис. 6. Отриманий елемент з серверу

Код

Script.js

```
let createdDropDowns = 0;
let createdLinks = 0;

function createDropDown(){
  const block = document.getElementById("b3");
  const target = document.getElementById("target");

  const newDropDown = document.createElement("div");
  newDropDown.className = "dropdown";
  newDropDown.id = `d${createdDropDowns}`;

  const newDropBtn = document.createElement("button");
  newDropBtn.className = "dropbtn";
  newDropBtn.textContent = "New";
  newDropBtn.id = `dbtn${createdDropDowns}`;
  newDropBtn.style.borderRight = "0";

  const newDropContent = document.createElement("div");
  newDropContent.className = "dropdown-content";
  newDropContent.id = `dcon${createdDropDowns}`;

  const newDropContentUp = document.createElement("div");
  newDropContentUp.className = "dropdown-content";

  const newDropContentUpEditBtn = document.createElement("button");
  newDropContentUpEditBtn.textContent = "Edit";
  newDropContentUpEditBtn.addEventListener('click', () => {
    openEditPopup(newDropBtn.id, false);
  });

  const newDropContentUpDeleteBtn = document.createElement("button");
  newDropContentUpDeleteBtn.textContent = "-";
  newDropContentUpDeleteBtn.addEventListener('click', () => {
    deleteElement(newDropDown.id);
  });

  const newDropContentBtn = document.createElement("button");
  newDropContentBtn.textContent = "+";
  newDropContentBtn.id = `dconbtn${createdDropDowns}`;
  newDropContentBtn.addEventListener('click', () => {
    createDropDownContentLink(newDropContent.id, newDropContentBtn.id);
  });

  newDropContent.appendChild(newDropContentBtn);

  newDropContentUp.appendChild(newDropContentUpDeleteBtn);
  newDropContentUp.appendChild(newDropContentUpEditBtn);

  newDropDown.appendChild(newDropContentUp);
  newDropDown.appendChild(newDropBtn);
  newDropDown.appendChild(newDropContent);

  block.insertBefore(newDropDown, target);
}
```



```

        createdDropDowns++;
    }

function createDropDownContentLink(blockID, targetID){
    const block = document.getElementById(blockID);
    const target = document.getElementById(targetID);

    const newDropContentLink = document.createElement("div");
    newDropContentLink.className = "dropdown-content-link";
    newDropContentLink.style.display = "flex";
    newDropContentLink.id = `l${createdLinks}`;

    const newLink = document.createElement("a");
    newLink.textContent = "newLink";
    newLink.target = "_blank";
    newLink.id = `a${createdLinks}`;

    const newEditLinkBtn = document.createElement("button");
    newEditLinkBtn.textContent = "Edit";
    newEditLinkBtn.addEventListener('click', () => {
        openEditPopup(newLink.id, true);
    });

    const newDeleteLinkBtn = document.createElement("button");
    newDeleteLinkBtn.textContent = "-";
    newDeleteLinkBtn.addEventListener('click', () => {
        deleteElement(newDropContentLink.id);
    });

    newDropContentLink.appendChild(newLink);
    newDropContentLink.appendChild(newEditLinkBtn);
    newDropContentLink.appendChild(newDeleteLinkBtn);

    block.insertBefore(newDropContentLink, target);
    createdLinks++;
}

function openEditPopup(elementID, isLink){
    const popup = document.getElementById("popup");
    const overlay = document.getElementById("blurOverlay");
    const buttonClose = document.getElementById("closePopup");
    const buttonSave = document.getElementById("saveBtnPopup");
    const element = document.getElementById(elementID);
    const textArea = document.getElementById("inputText");
    const textAreaURL = document.getElementById("inputTextURL");
    const textAreaLabelURL = document.querySelector(`label[for="inputTextURL"]`);

    popup.classList.add("active");
    overlay.classList.add("active");

    if (isLink) {
        textAreaURL.style.display = "block";
        textAreaLabelURL.style.display = "block";
    }
    else {
        textAreaURL.style.display = "none";
        textAreaLabelURL.style.display = "none";
    }
}

```

```

    }

    function closePopup() {
        popup.classList.remove("active");
        overlay.classList.remove("active");
    }

    buttonClose.onclick = closePopup;

    buttonSave.onclick = edit;

    function edit() {
        const input = textArea.value.trim();
        element.textContent = input;
        if (isLink) {
            const url = textAreaURL.value.trim();
            element.href = url;
        }
    }
}

function deleteElement(elementID){
    const element = document.getElementById(elementID);
    element.remove();
}

function collectDropdownData() {
    const dropdowns = document.querySelectorAll(".dropdown");
    const data = [];

    dropdowns.forEach((dropdown, index) => {
        if (index === dropdowns.length - 1) {
            return;
        }

        const links = [];

        dropdown.querySelectorAll(".dropdown-content-link").forEach(linkBlock => {
            const linkElement = linkBlock.querySelector("a");
            if (linkElement) {
                links.push({
                    href: linkElement.href,
                    text: linkElement.textContent
                });
            }
        });

        const dropbtn = dropdown.querySelector(".dropbtn");
        const dropdownName = dropbtn ? dropbtn.textContent : `dropdown-${index}`;

        data.push({
            dropdownName: dropdownName,
            links: links
        });
    });

    return data;
}

```

```

}

function sendDataToServer() {
  const data = collectDropdownData();

  fetch('index.php', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify(data)
  }).then(response => {
    if (response.ok) {
      console.log('Data successfully sent to server');
    } else {
      console.error('Error sending data');
    }
  });
}

```

Script1.js

```

function loadDropdowns() {
  fetch('dropdowns.json')
    .then(response => {
      if (!response.ok) {
        throw new Error(`HTTP error! status: ${response.status}`);
      }
      return response.json();
    })
    .then(data => {
      renderDropdowns(data);
      console.log("loaded");
    })
    .catch(error => {
      console.error('Error loading JSON:', error);
    });
}

function renderDropdowns(data) {
  const container = document.getElementById('b3');
  container.innerHTML = "";

  data.forEach(dropdown => {
    const dropdownDiv = document.createElement('div');
    dropdownDiv.classList.add('dropdown');

    const button = document.createElement('button');
    button.classList.add('dropbtn');
    button.textContent = dropdown.dropdownName;

    const dropdownContent = document.createElement('div');
    dropdownContent.classList.add('dropdown-content');

    dropdown.links.forEach(link => {
      const anchor = document.createElement('a');

```

```

        anchor.href = link.href;
        anchor.target = '_blank';
        anchor.textContent = link.text;

        dropdownContent.appendChild(anchor);
    });

    dropdownDiv.appendChild(button);
    dropdownDiv.appendChild(dropdownContent);

    container.appendChild(dropdownDiv);
});
}

document.addEventListener('DOMContentLoaded', () => {
    loadDropdowns();

    setInterval(loadDropdowns, 10000);
});

```

Index.php

```

<?php
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $jsonData = file_get_contents('php://input');

    if ($jsonData) {
        file_put_contents('dropdowns.json', $jsonData);

        header('Content-Type: application/json');
        echo json_encode(['status' => 'success']);
    } else {
        http_response_code(400);
        echo json_encode(['status' => 'error', 'message' => 'Invalid data']);
    }
}
?>

```

Посилання

–[Репозиторій](#)

–[Веб сторінка](#)

ВИСНОВОК

У ході виконання лабораторної роботи було досліджено принципи організації бекенду і фронтенду із застосуванням Javascript та PHP. Основна увага приділялася взаємодії між клієнтською частиною (фронтендом) та серверною частиною (бекендом) у веб-додатках, що функціонують у асинхронному режимі.

На першому етапі було створено дві веб-сторінки на основі попередньої роботи. Перша сторінка забезпечувала можливість користувачам динамічно створювати об'єкти (набори об'єктів) за допомогою Javascript та елементів форми. Було реалізовано механізми вибору кількості, порядку та контенту елементів, що дає змогу гнучко формувати структуру даних.

На другому етапі за допомогою Javascript та PHP організовано збереження об'єктів на сервері в асинхронному режимі. Було реалізовано механізми передачі даних на сервер із подальшим збереженням у вибраному форматі (файл або база даних). Це дало змогу ознайомитися з особливостями серверної обробки запитів та збереження даних.

Третій етап передбачав розробку другої веб-сторінки, яка отримувала дані з сервера та генерувала HTML, CSS і JS-код для відображення функціонального об'єкта. Було реалізовано періодичний асинхронний контроль змін даних на сервері з автоматичним оновленням відображуваних об'єктів. Такий підхід підвищив інтерактивність веб-додатка та дозволив відобразити реальну динаміку даних.

Отримані знання та практичні навички є важливими для подальшої розробки складних веб-додатків і забезпечення їх функціональності на високому рівні.