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 №4 з дисципліни «Програмування веб-застосувань»

«Розробка поштової та мобільної версії веб-сторінки» Варіант <u>13</u>

Виконав <u>ІП-33 Хребтань Максим Віталійович</u>

(шифр, прізвище, ім'я, по батькові)

Перевірив Очеретяний Олександр Костянтинович

(прізвище, ім'я, по батькові)

3MICT

ЛАБОРАТОРНА РОБОТА №5	3
Маніпуляції з HTML та CSS за допомогою Javascript	3
ХІД РОБОТИ	5
Посилання	14
ВИСНОВОК	

ЛАБОРАТОРНА РОБОТА №5

Маніпуляції з HTML та CSS за допомогою Javascript

Мета роботи: Ознайомлення із скриптовою мовою програмування Javascript на прикладі роботи із веб-сторінкою.

Завдання:

- 1. У вас вже є обраний із 16 варіантів власний варіант виконання роботи відповідно до вашого порядкового номера у списку групи, подальші маніпуляції виконуйте зі зверстаною головною сторінкою веб-сайта з л.р.№3.
- 2. За допомогою мови програмування Javascript виконайте вказані обчислення та проведіть необхідні маніпуляції з елементами веб-сторінки згідно з вашим варіантом, зберігаючи контент веб-сторінки, з урахуванням позиціювання та візуалізації окремих елементів веб-сторінки.
- 3. Для запису скриптів JS використовуйте будь-які відомі вам способи.
- 4. Переконайтеся, що функціональність скриптів збереглася для мобільної версії головної веб-сторінки. Якщо окрема функціональність порушена запропонуйте альтернативне рішення.
- 5. Виконане завдання опублікуйте на Github Pages (або обраному вами хостінгу) і продемонструйте його роботу.

Варіант 13

- 1. Поміняйте місцями тексти, позначені «1» та «6».
- 2. Напишіть функцію, яка обчислює площу п'ятикутника, беручи необхідні значення із відповідних змінних у скрипті, і виводить отриманий результат в кінці контенту в блоці «3».
- 3. Напишіть скрипт, який визначає можливість побудови трикутника із заданими довжинами сторін, беручи необхідні значення із форми в блоці «3», а отриманий результат виводить за допомогою діалогового вікна і зберігає в cookies, причому:
 - а) при оновленні веб-сторінки за допомогою діалогу виводиться інформація, збережена в cookies, із повідомленням, що після натискання кнопки «ОК» відбудеться видалення

даних із cookies, і не виводиться згадана вище форма;

- б) при натисканні кнопки «ОК» відповідні cookies видаляються, і виводиться наступне діалогове вікно із повідомленням про видалення cookies, а натискання кнопки «ОК» перезавантажує веб-сторінку з початковим станом із наявною формою для введення даних.
- 4. Напишіть скрипт, який при настанні події blur встановлює верхній регістр перших літер усіх слів у блоці «4» при встановленні користувачем відповідної галочки у формі і зберігає відповідне значення в localStorage броузера так, щоб при наступному відкриванні веб-сторінки властивість верхнього регістру перших літер усіх слів у блоці «4» встановлювалась із збереженого значення в localStorage.
- 5. Напишіть скрипт редагування тексту HTML-контейнерів:
- а) необхідні елементи форми появляються у відповідних номерних блоках (1..6) внаслідок подвійного кліку на елементі нумерованого списку (списку контейнерів) в блоці «3» (додайте такий список у блок «3» в кінці його вмісту);
- б) можна редагувати тільки контейнери із текстовими вузлами; наявні в них дочірні контейнери/теги не редагуються і не видаляються із структури;
- в) після текстового поля розміщується кнопка збереження зміненого тексту в localStorage і заміщення початкового тексту контейнера;
- г) після текстового поля розміщується кнопка видалення нового тексту із localStorage і відображення в контейнері початкового тексту без перезавантаження веб-сторінки.

ХІД РОБОТИ

Перший скрипт:

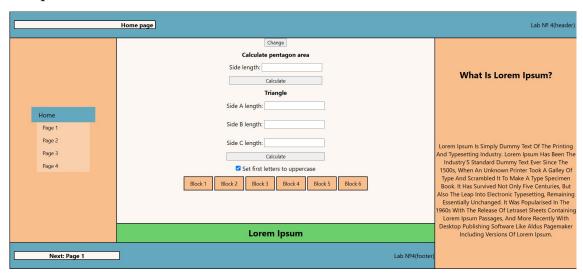


Рис. 1. Веб-сторінка до виконання

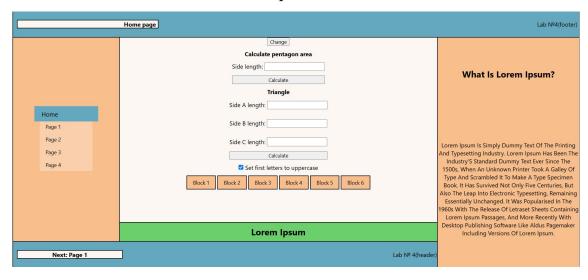


Рис. 2. Веб-сторінка після виконання (тексти змінено)

Другий скрипт:

Side length: 52	
Side length. 32	
Result: 4652.17	

Рис. 3. Для введеного значення сторони розраховано площу

Третій скрипт:

127.0.0.1:5501 says Possible	ОК
Calculate pentago	on area
Side length: 52	
Result: 4652.17	
Calculate	
Triangle	
Side A length: 3	
Side B length: 5	
Side C length: 6	
Calculate	

Рис. 4. Для введених значень сторін трикутника визначено можливість його побудови

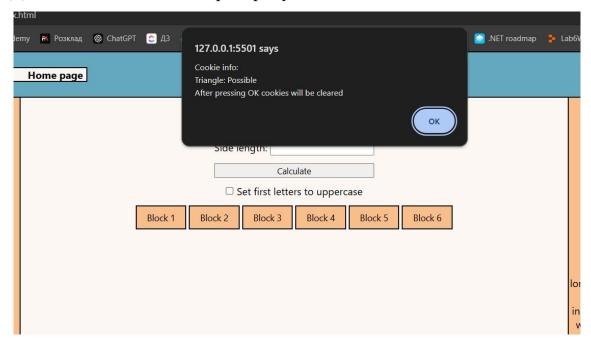


Рис. 5. Діалогове вікно після перезавантаження сторінки

Четвертий скрипт:

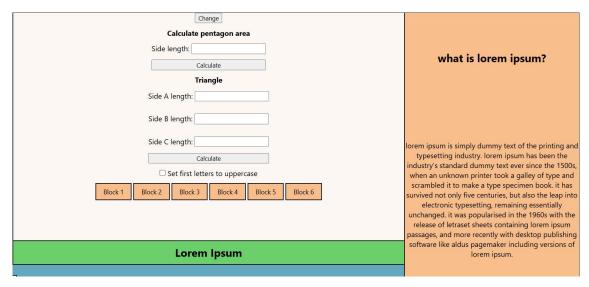


Рис. 6. Галочку не встановлено

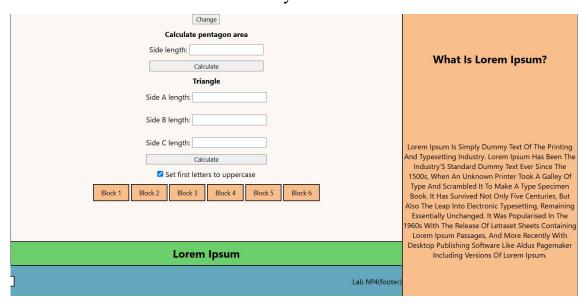


Рис. 7. Галочку встановлено та сталась подія blur

П'ятий скрипт:



Рис. 8. Викликано форму для додання тексту до блоку 1



Рис. 9. Блок 1 після збереження змін



Рис. 10. Блок 1 після видалення змін

Код усіх скриптів

```
function changeText(){
          let textH = document.getElementById("headTxt").innerHTML;
          let textF = document.getElementById("footTxt").innerHTML;
          document.getElementById("headTxt").innerHTML = textF;
          document.getElementById("footTxt").innerHTML = textH;
      }
      function calcPentagonArea(){
          let sideLength = parseFloat(document.getElementById("sideLength").value);
          if (isNaN(sideLength) || sideLength <= 0) {</pre>
              alert("Enter a valid value for the side length");
              return;
          }
          let area = (1 / 4) * Math.sqrt(5 * (5 + 2 * Math.sqrt(5))) * Math.pow(sideLength, 2);
          document.getElementById("resultPentagon").innerHTML = "Result: " + area.toFixed(2);
      }
      function checkTriangle(){
          const sides =
          [parseFloat(document.getElementById("sideA").value),
parseFloat(document.getElementById("sideB").value),
parseFloat(document.getElementById("sideC").value)];
          let m = sides.length;
          for (let i = 0; i < m; i++) {
              if (isNaN(sides[i]) || sides[i] <= 0) {</pre>
                  let message = "";
                  switch (i) {
                      case 0:
                          message = "Enter a valid value for the side A length";
                          break;
                      case 1:
                          message = "Enter a valid value for the side B length";
```

```
break;
                      case 2:
                          message = "Enter a valid value for the side C length";
                          break;
                      default:
                          break;
                  alert(message);
                  return;
              }
          }
          let result = "";
          sides[0] + sides[1] > sides[2] && sides[0] + sides[2] > sides[1] && sides[1] + sides[2] >
sides[0] ? result = "Possible" : result = "Not posssible";
          setCookie(result);
          alert(result);
      }
      function setCookie(cvalue) {
          document.cookie = "triangle=Triangle: " + cvalue;
      }
      window.onload = checkCookie;
      function checkCookie() {
          const cookieValue = document.cookie.split("; ").find((row) =>
row.startsWith("triangle="))?.split("=")[1];
          let message = "Cookie info:\n" + cookieValue + "\n" + "After pressing OK cookies will be
cleared";
          if (cookieValue !== undefined) {
              alert(message);
              document.cookie = "triangle=; expires= 01 Jan 1999 00:00:00 UTC;"
              alert("Cookies cleared!");
              location.reload();
          }
          else {
              let x = document.getElementById("formTriangle");
              x.style.display = "flex";
              x.style.flexDirection = "column";
          }
      }
      document.addEventListener("DOMContentLoaded", function() {
          const capitalizeCheckbox = document.getElementById("capitalizeCheckbox");
          const textBlock = document.getElementById("block4");
          applyInitialCapitalization();
```

```
function applyInitialCapitalization() {
        if (localStorage.getItem("capitalizeEnabled") === "true") {
            capitalizeCheckbox.checked = true;
            capitalizeFirstLetters(textBlock);
        } else {
            capitalizeCheckbox.checked = false;
            lowercaseFirstLetters(textBlock);
        }
    }
});
function handleBlurEvent(capitalizeCheckbox) {
    if (capitalizeCheckbox.checked) {
        capitalizeFirstLetters(document.getElementById("block4"));
        localStorage.setItem("capitalizeEnabled", "true");
    } else {
        lowercaseFirstLetters(document.getElementById("block4"));
        localStorage.setItem("capitalizeEnabled", "false");
    }
}
function capitalizeFirstLetters(element) {
    element.querySelectorAll("*").forEach(child => {
        if (child.nodeType === Node.ELEMENT_NODE) {
            child.innerHTML = child.innerHTML.replace(/\b\w/g, char => char.toUpperCase());
        }
    });
}
function lowercaseFirstLetters(element) {
    element.querySelectorAll("*").forEach(child => {
        if (child.nodeType === Node.ELEMENT NODE) {
            child.innerHTML = child.innerHTML.toLowerCase();
        }
    });
}
_id = undefined;
textNodes = undefined;
block = undefined;
function openForm(id) {
    const popup = document.getElementById("popup");
    const overlay = document.getElementById("blurOverlay");
    const textArea = document.getElementById("inputText");
    popup.classList.add("active");
    overlay.classList.add("active");
    _{id} = id;
```

```
const editedText = localStorage.getItem(_id + "t");
    if (editedText != null) {
        textArea.value = editedText;
    }
    else {
        textArea.value = "";
    }
    block = getBlock();
    textNodes = Array.from(block.childNodes).filter(node =>
        node.nodeType === Node.TEXT_NODE && node.nodeValue.trim() !== ""
    );
    checkEditedText();
}
function closePopup() {
    const popup = document.getElementById("popup");
    const overlay = document.getElementById("blurOverlay");
    popup.classList.remove("active");
    overlay.classList.remove("active");
}
function checkEditedText() {
    const buttonDelete = document.getElementById("deleteBtnPopup");
    const prevText = localStorage.getItem(_id);
    buttonDelete.disabled = prevText == null;
}
function saveText() {
    const textArea = document.getElementById("inputText");
    if (!textNodes[0]){
        const newTextNode = document.createTextNode("");
        block.appendChild(newTextNode);
        textNodes[0] = newTextNode;
    }
    const input = textArea.value.trim();
    if (localStorage.getItem( id) == null) {
        localStorage.setItem(_id, textNodes[0].nodeValue);
    localStorage.setItem( id + "t", input);
    textNodes[0].nodeValue = input;
    checkEditedText();
}
```

```
function deleteText() {
    const prevText = localStorage.getItem(_id);
    textNodes[0].nodeValue = prevText;
    localStorage.removeItem(_id);
    localStorage.removeItem(_id + "t");
    checkEditedText();
}
function getBlock() {
    let block = undefined;
    switch ( id) {
        case "b1":
            block = document.querySelector(".header");
            break:
        case "b2":
            block = document.querySelector(".block2");
        case "b3":
            block = document.querySelector(".block3");
            break;
        case "b4":
            block = document.querySelector(".block4");
            break;
        case "b5":
            block = document.querySelector(".block5");
            break:
        case "b6":
            block = document.querySelector(".footer");
            break;
        default:
            break;
    }
    return block;
}
window.addEventListener('unload', function () {
    const keysToKeep = ["capitalizeEnabled"];
    clearLocalStorageExcept(keysToKeep);
});
function clearLocalStorageExcept(keepKeys) {
    const savedItems = {};
    keepKeys.forEach(key => {
        const value = localStorage.getItem(key);
        if (value !== null) {
            savedItems[key] = value;
        }
    });
```

```
localStorage.clear();

for (const key in savedItems) {
    localStorage.setItem(key, savedItems[key]);
}
```

Посилання

- -<u>Репозиторій</u>
- -Веб сторінка

ВИСНОВОК

Під час виконання лабораторної роботи я ознайомився з базовими і просунутими можливостями мови програмування JavaScript для маніпуляції елементами веб-сторінки. У ході роботи було реалізовано завдання, що стосуються зміни контенту, обчислень, збереження даних у cookies та localStorage, а також забезпечення інтерактивності веб-сторінки.

Ця лабораторна робота допомогла краще зрозуміти методи роботи з DOM, подіями, cookies та localStorage. Робота показала можливості JavaScript у динамічному управлінні контентом, збереженні даних та інтерактивності вебсторінок. Використання таких технік є важливим для створення сучасних адаптивних веб-додатків.