Лабораторна робота №7

Робота з DOM та BOM у JavaScript. Створення системи реєстрації на захід

Мета:

Навчитися працювати з DOM та BOM, використовувати методи для маніпуляції елементами сторінки, обробляти події, здійснювати валідацію форми з відображенням помилок, зберігати дані в Local Storage, фіксувати дату і час реєстрації та позначати відвідування заходу.

Завдання:

Дата і час:

 Додати на верху сторінки блок з поточною датою і часом, котрі мають оновлюватися кожної секунди.

2. Створення HTML-сторінки з формами:

- Реалізувати HTML-форму для введення імені адміністратора.
- Реалізувати HTML-форму для реєстрації учасників, що включає поля: Ім'я,
 Прізвище, Email та Телефон.

3. Збереження даних адміністратора в Local Storage:

- 。 Реалізувати функціонал, який зберігає ім'я адміністратора в Local Storage.
- Під час завантаження сторінки перевіряти наявність даних адміністратора в Local Storage та заповнювати відповідне поле, якщо дані існують.

4. Реєстрація учасників:

- о Після натискання кнопки "Реєстрація" додавати дані учасника до списку на сторінці разом із позначенням дати й часу реєстрації.
- 。 Очищати поля форми після успішної реєстрації.

5. Валідація форми учасника:

- 。 Перевіряти, щоб усі поля форми були заповнені.
- Поля "Ім'я" та "Прізвище" мають приймати лише українські літери та дефіс;
 для цього використовуйте регулярні вирази.
- Використовувати регулярні вирази для перевірки правильності введення Етаіl та Телефону.

- □ Поля "Ім'я" та "Прізвище" мають містити щонайменше 3 символи, а номер телефону — не менше 7 символів.
- Виводити повідомлення про помилки під полями форми, якщо дані не пройшли перевірку.

6. Видалення учасника зі списку:

Додати можливість видалення кожного учасника з відображеного списку.

7. Позначення відвідування заходу:

- 。 Додати кнопки "Відвідано" і "Не відвідано" біля кожного учасника.
- о Під час натискання "Відвідано" змінювати стиль елемента (колір або фон блоку) для позначення відвідування.
- о Під час натискання "Не відвідано" змінювати стиль елемента (колір або фон блоку) для позначення, що учасник не відвідав захід.

8. Мінімальна стилізація сторінки:

- Зробити базову стилізації сторінки включно з кнопками, формою та списком учасників.
- 。 Задати відступи, розмір шрифту та вирівнювання для форми.
- Додати стилі для помилок валідації, наприклад колір тексту помилки і колір input поля.
- Змінити фон або колір тексту для візуального позначення учасників, які відвідали захід або не відвідали.

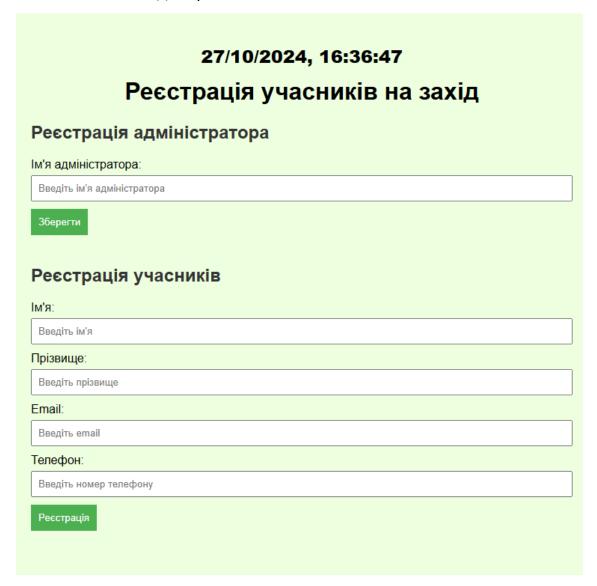
Очікуваний результат:

Веб-сторінка повинна в реальному часі оновлювати дату та час, дозволяючи адміністратору зберігати своє ім'я, реєструвати учасників на захід з валідацією даних, а також відмічати їх відвідування чи відсутність і видаляти зі списку.

Приклад реалізації

Наступна інформація наводиться лише для ілюстрації. Ваша реалізація може відрізнятися як у деталях, так і в загальному підході.

Початковий вигляд сторінки:



дата і час оновлюються автоматично кожної секунди.

Приклад валідації:

27/10/2024, 16:38:56 Реєстрація учасників на захід

Реєстрація учасників на захід	
Реєстрація адміністратора	
Ім'я адміністратора:	
Андрій	
Зберегти Реєстрація учасників	
Ім'я:	
Введіть ім'я	
Ім'я повинно містити тільки українські літери (3+ символи).	
Прізвище:	
Введіть прізвище	
Прізвище повинно містити тільки українські літери (3+ символи).	
Email:	
Введіть email	
Некоректний Email.	
Телефон:	
Введіть номер телефону	
Некоректний номер телефону (мінімум 7 цифр).	
Реєстрація	

Приклад заповнення форми:

27/10/2024, 16:40:57

21/10/2024, 10:40:57	
Реєстрація учасників на захід	
Реєстрація адміністратора	
Ім'я адміністратора:	
Андрій	
Зберегти	
December in the second	
Реєстрація учасників	
lm'я:	
Назар	
Прізвище:	
Жменька	
Email:	
zhmenka@gmail.com	
Телефон:	
0442370255	
Реєстрація	

27/10/2024, 16:43:26		
Реєстрація учасників на захід		
Реєстрація адміністратора		
м'я адміністратора:		
Андрій		
Зберегти		
Реєстрація учасників		
м'я:		
Введіть ім'я		
Трізвище:		
Введіть прізвище		
Email:		
Введіть email		
Гелефон:		
Введіть номер телефону		
Список учасників		
Назар Жменька zhmenka@gmail.com 0442370255 27/10/2024, 16:41:21 Відвідано Не відвідано Видалити		
Андрій Брунько brunko@yahoo.com +380960000000 27/10/2024, 16:42:15 Відвідано Не відвідано Видалити		
Анна Крупа a.krupa@ukr.net 2394724 27/10/2024, 16:43:21 Відвідано Не відвідано Видалити		

Приклад позначення відвідування заходу учасниками (відвіано/не відвідано):

27/10/2024, 16:49:19 Реєстрація учасників на захід Реєстрація адміністратора Ім'я адміністратора: Андрій Реєстрація учасників Ім'я: Введіть ім'я Прізвище: Введіть прізвище Email: Введіть email Телефон: Введіть номер телефону Реєстрація Список учасників Назар Жменька | zhmenka@gmail.com | 0442370255 | 27/10/2024, 16:41:21 Не відвідано Відвідано Видалити Андрій Брунько | brunko@yahoo.com | +380960000000 | 27/10/2024, 16:42:15 Відвідано Не відвідано Видалити Анна Крупа | a.krupa@ukr.net | 2394724 | 27/10/2024, 16:43:21

Відвідано

Не відвідано

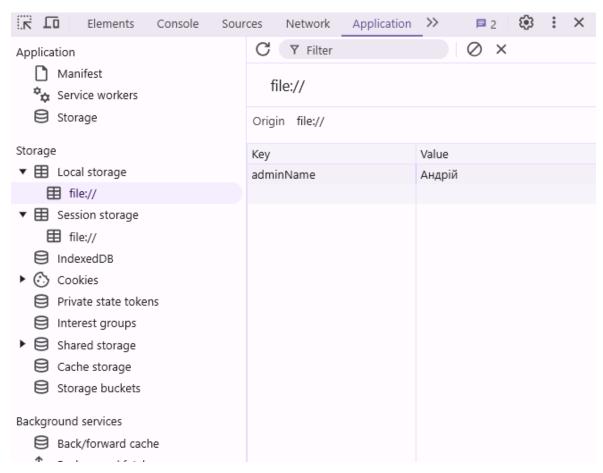
Видалити

27/10/2024, 16:49:55

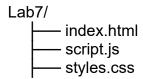
Реєстрація адміністратора	
Ім'я адміністратора:	
Андрій	
Зберегти	
Реєстрація учасників	
Ім'я:	
Введіть ім'я	
Прізвище:	
Введіть прізвище	
Email:	
Введіть email	
Телефон:	
Введіть номер телефону	
Список учасників	
Назар Жменька zhmenka@gmail.com 0442370255 27/10/2024, 16:41:21	
Відвідано Не відвідано Видалити	
Анна Крупа a.krupa@ukr.net 2394724 27/10/2024, 16:43:21 Відвідано Не відвідано Видалити	

Приклад базової структури html сторінки:

Приклад збереження даних у Local Storage



Структура архіву:



Матеріали та довідники:

- ✓ w3schools.com JavaScript (англійською мовою)
- ✓ w3schoolsua.github.io JavaScript (українською мовою)
- ✓ MDN Web Docs JavaScript (англійською мовою)
- ✓ regex101.com
- √ w3schools.com JavaScript RegExp Reference

Рекомендовані редактори коду:

- ✓ VSCode
- ✓ PSPad
- ✓ Notepad++
- ✓ Sublime Text (платний)
- ✓ WebStorm, PHPStorm, PyCharm (платний, <u>безкоштовна ліцензія для навчання</u>)

Вимоги до звіту:

Після виконання завдання, студент повинен надати звіт, який включає:

1. Короткий опис виконаної роботи:

- Опис мети лабораторної роботи.
- Короткий опис завдання та його кроків виконання.
- 2. Скріншоти коду з коментарями та результатів виконання завдання.

3. Висновок:

- Короткий підсумок виконаної роботи.
- Висновки щодо досягнення мети лабораторної роботи.
- Відповіді на питання, які могли виникнути під час виконання роботи.

4. Архів з файлами лабораторної роботи:

- Повинен містити HTML, CSS та JS файли.
- Архівуйте файли у форматі ZIP або RAR.

Контрольні запитання для самоконтролю:

Питання по DOM

1. Структура DOM:

- Що таке DOM, і як він представляє HTML-документ?
- Яка структура DOM-дерева, і як елементи відображаються у вигляді вузлів?
- Які типи вузлів є в DOM?
- Як DOM взаємодіє з JavaScript?
- Як браузери обробляють і рендерять DOM?

2. Доступ до елементів DOM:

- Які методи використовуються для пошуку елементів на сторінці?
- Яка різниця між методами querySelector та getElementById?
- Як можна отримати або змінити текстовий вміст і атрибути елементів?
- Як отримати колекцію елементів за допомогою getElementsByClassName або getElementsByTagName?
- Як браузери обробляють і рендерять DOM?

3. Зміна структури DOM:

- Як можна додати новий елемент до DOM?
- Які методи використовуються для видалення, заміни або клонування елементів?
- Як змінити атрибути, класи або стилі елемента?
- Як використовувати методи insertBefore та appendChild?
- Як використовувати методи createElement та createTextNode?

4. **Події в DOM:**

- Що таке подія в контексті DOM, і як можна додати обробник події до елемента?
- Які основні події підтримує DOM?

- У чому різниця між addEventListener і використанням HTML-атрибутів для прив'язки подій?
- Як використовувати методи preventDefault та stopPropagation?

5. Маніпуляція класами та стилями:

- Як додавати, видаляти або перемикати CSS-класи за допомогою JavaScript?
- Як можна динамічно змінювати стилі елементів?
- Що таке classList, і які методи він містить?
- Як змінювати стилі елементів за допомогою властивості style?

6. Навігація по DOM-дереву:

- Як отримати доступ до батьківських, дочірніх або сусідніх елементів?
- У чому різниця між children та childNodes?
- Як працюють властивості parentNode, firstChild, nextSibling?
- Як використовувати методи closest та matches для навігації по DOM?
- Як використовувати методи parentElement, previousElementSibling та nextElementSibling?

Питання по ВОМ

1. Огляд ВОМ:

- Що таке ВОМ, і чим він відрізняється від DOM?
- Які об'єкти належать до ВОМ?
- Як ВОМ взаємодіє з браузером?
- Як використовувати об'єкт console для налагодження?

2. Ob'εκτ window:

- Що таке об'єкт window, і чому його можна використовувати неявно?
- Які основні властивості та методи має window?
- Як window представляє глобальний контекст у JavaScript?

Як використовувати методи open та close для роботи з вікнами?

3. Poбота з location:

- Що таке об'єкт location, і як він представляє URL поточної сторінки?
- Як змінити URL сторінки за допомогою location?
- Які властивості має location для отримання протоколу, домену, хешу тощо?
- Як використовувати методи assign, replace та reload?

4. Об'єкт navigator:

- Яку інформацію надає об'єкт navigator про браузер та систему?
- Як отримати інформацію про мову браузера або підтримувані типи файлів?
- У чому можуть полягати обмеження при використанні navigator для ідентифікації браузера?
- Як використовувати navigator для перевірки підтримки сервіс-воркерів?

5. **Об'єкт history:**

- Що таке об'єкт history, і як він працює з історією браузера?
- Які методи history використовуються для навігації
- Як працюють методи back, forward, go?
- Чи є обмеження у використанні history для безпеки та конфіденційності користувачів?
- Як використовувати методи pushState та replaceState для маніпуляції історією?

6. Таймери та анімації:

- Як працюють методи setTimeout i setInterval?
- Як можна зупинити виконання таймера, запущеного за допомогою setTimeout aбо setInterval?
- Як часто варто використовувати таймери у додатках, щоб зберігати продуктивність?
- Як використовувати requestAnimationFrame для створення анімацій?

7. Storage, Session Storage та Cookies:

- Що таке Local Storage, Session Storage та Cookies, і які основні відмінності між ними?
- Які дані зберігаються в Local Storage, Session Storage та Cookies?
- Як довго зберігаються дані в Local Storage, Session Storage та Cookies?
- Які обмеження на обсяг даних існують для Local Storage, Session Storage та Cookies?
- Як доступ до даних у Local Storage, Session Storage та Cookies здійснюється з JavaScript?
- Які переваги та недоліки використання Local Storage порівняно з Session Storage та Cookies?
- У яких випадках краще використовувати Local Storage, а в яких Session Storage aбо Cookies?
- Як зберегти дані y Local Storage, Session Storage та Cookies за допомогою JavaScript?
- Як видалити дані з Local Storage, Session Storage та Cookies за допомогою JavaScript?
- Як дані, збережені в Cookies, можуть бути доступні як на стороні клієнта, так і на стороні сервера?