ЗАНЯТТЯ 14 РОБОТА З ДАНИМИ КОРИСТУВАЧА. ПРОДОВЖЕННЯ ФОРМ. ОГЛЯД ВІДПРАВКИ ДАНИХ НА СЕРВЕР

ТЕМИ

31. Отримання даних від користувача та робота з ними

- Форми вводу даних. Стандартні елементи.
- Валідація (перевірка) даних на боці клієнта.
- Збереження даних на боці клієнта - localstorage, session storage.
- Побудова логіки сторінки, орієнтованої на роботу з даними.

Практика:

- Створюємо форму входу на сайт та інтегруємо її у сторінку (на вибір ментора)

РЕКОМЕНДАЦІЇ ДО ПРОВЕДЕННЯ ЗАНЯТТЯ

1. **Обговорюємо разом домашнє завдання**
   - Бажано, щоб ментор мав свій варінат розв'язку.
   - Дивимось варіанти учасників. Якщо є підходящі варіанти, то дороблюємо разом з групою один-два з них (там, де є напрацювання, але щось не вийшло, і можна доробити досить швидко).
   - Якщо доцільно, демонструємо свій варіант, та обговорюємо його з групою.

2. **Надання користувачу можливості вводу даних**
   - Робимо мотиваційний вступ
     - Наголошуємо на тому, що майже кожен веб-сайт так чи інакше включає сценарії, де користувач повинен вводити інформацію (від логіну та паролю до форм зворотнього зв'язку)
     - Аргументуємо, що в більшості випадків, введені дані повинні відправлятися з клієнта (браузера) на сервер, та впливати на поведінку системи як на боці клієнта, так і на боці сервера (дані про замовлення товарів, профілі користувача, логін пароль для входу, відгуки, обрані фільтри…).
   - Описуємо дві частини задачі
     - дати можливість користувачу ввести дані
     - власне, відправка даних на сервер в зрозумілому для нього форматі
   - Розповідаємо про форми та експерементуємо разом з групою
     - Показуємо базовий синтаксис форми, приклад з текстовими полями вводу
     - Демонструємо стилізацію за допомогою bootstrap (також базово, без розбору всіх варіантів - просто щоб форма виглядала сучасно)
     - Додаємо кілька найбільш вживаних типів полів вводу, окрім текстових (select, password, checkbox…)

- Розповідаємо про поняття “відправки форми” - submit, і про необхідність задання дії (action) форми.
- Поверхнево згадуємо про метод відправки форми (method), та про різницю між GET та POST.
- Створюємо тестову сторінку - “приймач” - ту, на яку відправляємо нашу форму, використовуючи метод GET та атрибут action з відповідним url. Додаємо кнопку “відправити”.
- Показуємо, що при методі GET параметри форми потрапляють в адресний рядок браузера - додаються до посилання.
- Обговорюємо принципи, за якими це відбувається, пояснюємо атрибут name у полів вводу.
- Демонструємо, що відбувається, якщо ввести в поле текст українською, або ввести слова латиницею але з пробілом. Обговорюємо поняття url encoding.

- Пропонуємо групі разом з ментором розробити сторінку “входу на сайт” (логіну) - і “захищену сторінку” сайту, яка імітує перевірку імені користувача та паролю, виконуючи цю перевірку на боці клієнта. Приклад реалізованих сторінок:
  https://github.com/sergdeni33/WebDevDemoShop/blob/master/JsStart/login.html
  https://github.com/sergdeni33/WebDevDemoShop/blob/master/JsStart/login-result.html
  https://github.com/sergdeni33/WebDevDemoShop/blob/master/JsStart/js/check-login.js
  - Створюємо просту бутстрап-стилізовану форму з полями “ім’я користувача”, “пароль” та “запам’ятати мене” (checkbox). По натисненню кнопки “Увійти” переходимо на іншу нашу сторінку - “дані користувача”.
  - На сторінці результату реалізуємо скріпт, який відразу при її відкритті виймає з адресного рядка браузера передані параметри, “парсить” їх (попередньо зробивши urlDecode), звіряє ім’я та пароль з деякими тестовими, які можна прописати прямо в коді сторінки - наприклад, зробити там масив допустимих користувачів.
  - В залежності від “вірного” чи “невірного” логіну, або показуємо сторінку - щось на зразок “Вітаємо, Олексій!” - або ж перенаправляємо браузер назад на сторінку вводу даних (по ходу демонструємо роботу з об’єктом location).
  - Якщо вистачає часу, можна додати до форми також деяке поле вибору, і на сторінці результату відображати обране значення.
  - Якщо група достатньо сильна, можна реалізувати функціональність чекбоксу “Запам’ятати мене”, використовуючи localstorage (див. приклад за посиланням).

3. **Обговорюємо отриманий результат, намагаємось провокувати питання в учасників та обговорення.** Ще раз наголошуємо, що це була демонстрація технологій - і в реальності не можна зберігати та перевіряти логін-пароль на клієнті - при цьому принципи пердачі отримання параметрів і т.д. залишаються тими самими і в варіанті з реальною серверною частиною. Описуємо, як відбувається взаємодія зі справжнім сервером. Можна показати запити на якомусь прикладі, використовуючи консоль браузера або інструмент типу Fiddler.

ДОМАШНЄ ЗАВДАННЯ (ТЕОРІЯ)

- Курс з веб-розробки на W3Schools, тема “JavaScript Forms” (англ)
- Курс з веб-розробки на W3Schools, тема “HTML5 Web Storage” (англ)

- Mozilla Developer Network (MDN) «Your first HTML form» (англ)
- Mozilla Developer Network (MDN) «Form data validation» (англ)
- Mozilla Developer Network (MDN) «Web Storage API» (англ)
- Mozilla Developer Network (MDN) «Third party APIs» (англ)

ДОМАШНЄ ЗАВДАННЯ (ПРАКТИКА)

- Пропонуємо створити сторінку входу до нашого “магазину”, на якій користувач вводить логін пароль, та обирає валюту. Після цього він потрапляє на сторінку “каталогу плиткою” на основі попереднього завдання, де додаємо перевірку логіну та паролю. При невірному введенні - повертаємо його на сторінку входу. При вірному - відображаємо каталог та ціни в обраній валюті, і біля кожного товару кнопку “придбати”, а також відображаємо десь на сторінці “вітання” - “Ви зайшли як …”, та кнопку “Вийти”.
- Сторінка каталогу повинна, як і раніше, допускати перегляд і без логіну та паролю - але у такому випадку біля товарів не повинно бути кнопки “придбати”, ціна повинна бути в гривні, і на сторінці не повинно бути “вітання” та “вийти” - навпаки, повинне бути посилання “Увійти”, яке веде на сторінку вводу логіну та паролю.

ДОДАТКОВІ КОРИСНІ МАТЕРІАЛИ

- Навчальна платформа Devionity, курс Основы CSS, заняття «Пример из жизни: валидация формы при помощи jQuery» (рос)
- Навчальна платформа Devionity, курс Основы CSS, заняття «Пример из жизни: валидация формы при помощи JS» (рос)