**Лабораторная работа № 19-20**

**Хранение данных на клиенте (JSON+COOKIE+WEB STORAGE)**

**Вопросы для самоконтроля:**

**JSON**

1. Опишите понятия API, JSON, XML, FRONT-END, BACK-END, AJAX. Есть ли между ними связь?

* API (Application Programming Interface): Интерфейс, который определяет, как программные компоненты взаимодействуют друг с другом. API часто используют JSON или XML для обмена данными.
* JSON (JavaScript Object Notation): Легковесный формат обмена данными, основанный на структуре объектов JavaScript.
* XML (Extensible Markup Language): Более сложный формат данных, используемый для структурирования и передачи информации.
* Front-end: Клиентская часть приложения, отвечающая за отображение и взаимодействие с пользователем.
* Back-end: Серверная часть приложения, которая обрабатывает данные, выполняет логику и взаимодействует с базой данных.
* AJAX (Asynchronous JavaScript and XML): Методология асинхронного обмена данными между клиентом и сервером без перезагрузки страницы. AJAX часто использует JSON.

Связь: JSON — основной формат данных для обмена через API между Front-end и Back-end. AJAX служит инструментом для работы с этими данными в реальном времени.

1. JSON – что это? Принцип действия JSON. Зачем он нужен?

* JSON — это текстовый формат, используемый для сериализации данных (представления их в виде строки) и передачи между системами.
* Принцип действия:

1. Данные создаются (например, сервером или клиентом).
2. Преобразуются в строку JSON.
3. Передаются через сеть.
4. Распаковываются (десериализуются) для работы в другой системе.

* Зачем нужен: JSON облегчает обмен данными между разными платформами и языками программирования.

1. Синтаксис и структура JSON.

* JSON состоит из ключей (строки) и значений (строки, числа, массивы, объекты, логические значения или null).
* Пример структуры:

{

"name": "John",

"age": 30,

"isStudent": false,

"courses": ["Math", "Science"],

"address": {

"city": "New York",

"zipcode": "10001"

}

}

1. Как работать с JSON?

* На стороне клиента:
* Используются методы JavaScript: JSON.stringify() для преобразования объекта в строку и JSON.parse() для обратного преобразования.
* На стороне сервера:
* Серверы используют библиотеки для работы с JSON (например, json в Python, Jackson в Java).

1. Преимущества JSON.

* Читаемость и простота.
* Компактность по сравнению с XML.
* Поддержка всеми популярными языками программирования.
* Легкость работы с массивами и объектами.

1. XML и JSON? Что выбрать?

* JSON:
* Легче в использовании, особенно для веб-приложений.
* Подходит для большинства сценариев API.
* XML:
* Имеет больше возможностей (например, схемы валидации, комментарии).
* Предпочтителен в сложных системах или при необходимости строгой структуры данных.

**COOKIE и WEB STORAGE**

1. Укажите известные вам способы сохранения данных на клиенте?

* Cookie
* Local Storage
* Session Storage
* IndexedDB

1. Для чего предназначены cookie и Web storage? Какие преимущества и недостатки имеет каждый способ хранения?

* Cookie:
* Для сохранения данных, передаваемых между клиентом и сервером.
* Преимущества: автоматически отправляются на сервер.
* Недостатки: ограниченный объем (около 4 КБ), замедляют запросы.
* Web Storage:
* Local Storage: Данные сохраняются бессрочно.
* Session Storage: Данные сохраняются только в рамках текущей сессии.
* Преимущества: больший объем данных (до 5 МБ), не передаются на сервер.
* Недостатки: нельзя получить доступ с сервера.

1. Как установить и как получить значения куки? Приведите примеры.

* **Установка куки**:

document.cookie = "username=John; path=/; expires=Fri, 01 Jan 2025 12:00:00 UTC";

* **Получение куки**:

console.log(document.cookie); // "username=John"

1. Опишите назначение объектов Session storage и Local storage.

* Local Storage: Используется для длительного хранения данных (например, пользовательские настройки).
* Session Storage: Предназначен для данных, которые нужны только в рамках текущей сессии (например, временные токены).

1. Как установить и как получить значения из хранилища Web storage? Приведите примеры.

* **Local Storage**:

localStorage.setItem("theme", "dark");

console.log(localStorage.getItem("theme")); // "dark"

* **Session Storage**:

sessionStorage.setItem("cartItems", "5");

console.log(sessionStorage.getItem("cartItems")); // "5"

**Задание 1 JSON и корзина Интернет-магазина.**

По своей индивидуальной теме (макет берем из практики по web-программированию) необходимо подготовить JSON-файл с данными для обработки (не менее 10 записей для каталога товаров или услуг). Найти в сети Интернет примеры кода для реализации КОРЗИНЫ ИНТЕРНЕТ-МАГАЗИНА. Реализовать работу с каталогом товара и корзиной для вашего шаблона через использование JSON-файла.

**Код:**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Online Store</title>

  <style>

    body { font-family: Arial, sans-serif; margin: 20px; }

    .catalog, .cart { margin-top: 20px; }

    .product { border: 1px solid #ccc; padding: 10px; margin: 10px 0; }

    .cart-item { border-bottom: 1px solid #ccc; padding: 5px 0; }

  </style>

</head>

<body>

  <h1>Online Store</h1>

  <div class="catalog">

    <h2>Catalog</h2>

    <div id="product-list"></div>

  </div>

  <div class="cart">

    <h2>Shopping Cart</h2>

    <div id="cart-items"></div>

    <p>Total: $<span id="cart-total">0</span></p>

  </div>

  <script>

    // JSON-данные, встроенные в код

    const products = [

      { "id": 1, "name": "Laptop", "price": 800, "description": "High performance laptop" },

      { "id": 2, "name": "Smartphone", "price": 600, "description": "Latest model smartphone" },

      { "id": 3, "name": "Headphones", "price": 100, "description": "Noise-cancelling headphones" },

      { "id": 4, "name": "Keyboard", "price": 50, "description": "Mechanical keyboard" },

      { "id": 5, "name": "Mouse", "price": 30, "description": "Wireless mouse" },

      { "id": 6, "name": "Monitor", "price": 200, "description": "4K monitor" },

      { "id": 7, "name": "Tablet", "price": 400, "description": "10-inch tablet" },

      { "id": 8, "name": "Speaker", "price": 120, "description": "Bluetooth speaker" },

      { "id": 9, "name": "Printer", "price": 150, "description": "Laser printer" },

      { "id": 10, "name": "Router", "price": 90, "description": "High-speed router" }

    ];

    const productList = document.getElementById('product-list');

    const cartItems = document.getElementById('cart-items');

    const cartTotal = document.getElementById('cart-total');

    let cart = [];

    // Отображение каталога

    products.forEach(product => {

      const productDiv = document.createElement('div');

      productDiv.className = 'product';

      productDiv.innerHTML = `

        <h3>${product.name}</h3>

        <p>${product.description}</p>

        <p>Price: $${product.price}</p>

        <button onclick="addToCart(${product.id})">Add to Cart</button>

      `;

      productList.appendChild(productDiv);

    });

    // Функция добавления в корзину

    function addToCart(id) {

      const product = products.find(p => p.id === id);

      cart.push(product);

      updateCart();

    }

    // Обновление корзины

    function updateCart() {

      cartItems.innerHTML = '';

      let total = 0;

      cart.forEach((item, index) => {

        const cartItem = document.createElement('div');

        cartItem.className = 'cart-item';

        cartItem.innerHTML = `

          <p>${item.name} - $${item.price}</p>

          <button onclick="removeFromCart(${index})">Remove</button>

        `;

        cartItems.appendChild(cartItem);

        total += item.price;

      });

      cartTotal.textContent = total;

    }

    // Функция удаления из корзины

    function removeFromCart(index) {

      cart.splice(index, 1);

      updateCart();

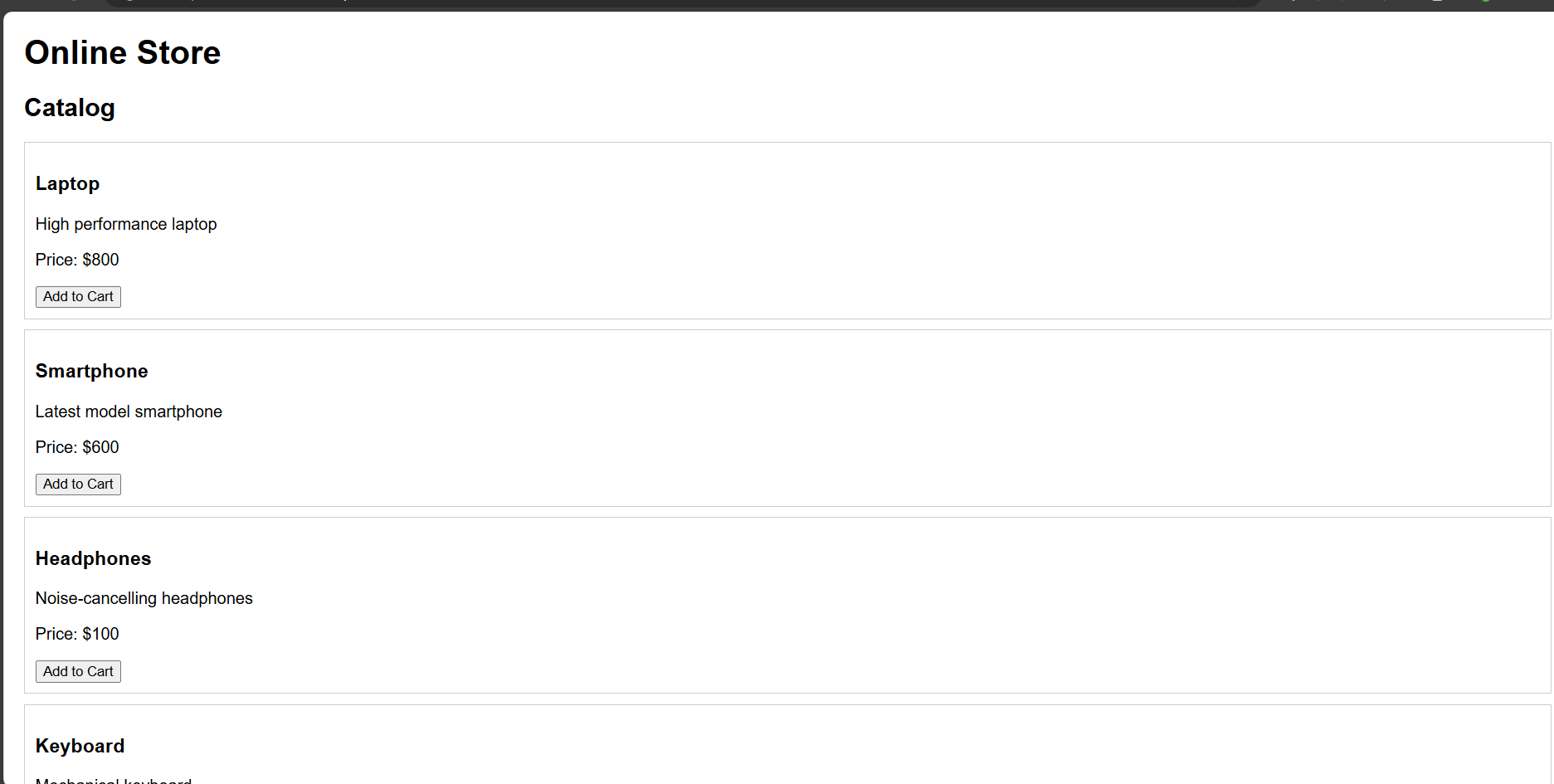
    }

  </script>

</body>

</html>

**Результат:**

**Задание 2**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Cookie Example</title>

</head>

<body>

  <h1>Save Your Data</h1>

  <form id="user-form">

    <label>Full Name: <input type="text" id="name"></label><br>

    <label>Email: <input type="email" id="email"></label><br>

    <label>Date of Birth: <input type="date" id="dob"></label><br>

    <label>Place of Birth: <input type="text" id="place"></label><br>

    <label>Hobbies: <input type="text" id="hobbies"></label><br>

    <button type="button" onclick="saveToCookies()">Save</button>

    <button type="button" onclick="clearCookies()">Clear Cookies</button>

  </form>

  <h2>Saved Data</h2>

  <div id="saved-data"></div>

  <script>

    function saveToCookies() {

      const name = document.getElementById('name').value;

      const email = document.getElementById('email').value;

      const dob = document.getElementById('dob').value;

      const place = document.getElementById('place').value;

      const hobbies = document.getElementById('hobbies').value;

      document.cookie = `name=${name}`;

      document.cookie = `email=${email}`;

      document.cookie = `dob=${dob}`;

      document.cookie = `place=${place}`;

      document.cookie = `hobbies=${hobbies}`;

      displayCookies();

    }

    function displayCookies() {

      const savedData = document.getElementById('saved-data');

      savedData.textContent = document.cookie;

    }

    function clearCookies() {

      document.cookie = "name=; expires=Thu, 01 Jan 1970 00:00:00 UTC;";

      document.cookie = "email=; expires=Thu, 01 Jan 1970 00:00:00 UTC;";

      document.cookie = "dob=; expires=Thu, 01 Jan 1970 00:00:00 UTC;";

      document.cookie = "place=; expires=Thu, 01 Jan 1970 00:00:00 UTC;";

      document.cookie = "hobbies=; expires=Thu, 01 Jan 1970 00:00:00 UTC;";

      displayCookies();

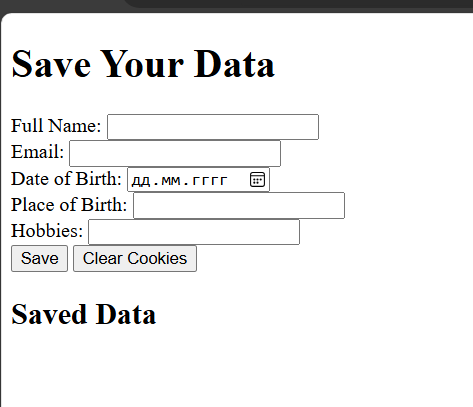
    }

    displayCookies();

  </script>

</body>

</html>

**Результат:**

**Задание 3**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Cookie Example</title>

</head>

<body>

  <h1>Save Your Data</h1>

  <form id="user-form">

    <label>Full Name: <input type="text" id="name"></label><br>

    <label>Email: <input type="email" id="email"></label><br>

    <label>Date of Birth: <input type="date" id="dob"></label><br>

    <label>Place of Birth: <input type="text" id="place"></label><br>

    <label>Hobbies: <input type="text" id="hobbies"></label><br>

    <button type="button" onclick="saveToCookies()">Save</button>

    <button type="button" onclick="clearCookies()">Clear Cookies</button>

  </form>

  <h2>Saved Data</h2>

  <div id="saved-data"></div>

  <script>

    function saveToLocalStorage() {

  const name = document.getElementById('name').value;

  const email = document.getElementById('email').value;

  const dob = document.getElementById('dob').value;

  const place = document.getElementById('place').value;

  const hobbies = document.getElementById('hobbies').value;

  localStorage.setItem('name', name);

  localStorage.setItem('email', email);

  localStorage.setItem('dob', dob);

  localStorage.setItem('place', place);

  localStorage.setItem('hobbies', hobbies);

  displayLocalStorage();

}

function displayLocalStorage() {

  const savedData = document.getElementById('saved-data');

  savedData.textContent = `

    Name: ${localStorage.getItem('name')},

    Email: ${localStorage.getItem('email')},

    DOB: ${localStorage.getItem('dob')},

    Place: ${localStorage.getItem('place')},

    Hobbies: ${localStorage.getItem('hobbies')}

  `;

}

function clearLocalStorage() {

  localStorage.clear();

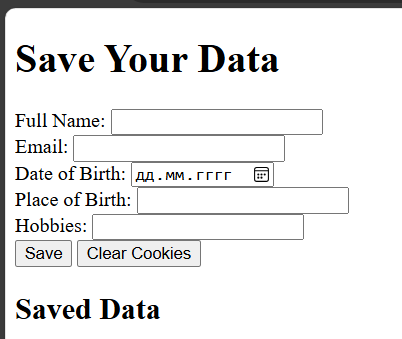
  displayLocalStorage();

}

  </script>

</body>

</html>

**Результат:**