

Практическая работа №2. Создание клиентской WEB страницы для приложения «Моя ГИБДД»

Single Page Application (SPA) - это одностраничное приложение (англ. «сингл пейдж аппликейшен, SPA) — это веб-приложение или веб-сайт, использующий единственный HTML-документ как оболочку для всех веб-страниц и организующий взаимодействие с пользователем через динамически подгружаемые HTML, CSS, JavaScript, обычно посредством AJAX.

Одностраничники похожи на «native» (англ) или иным словом «родные»(рус) приложения для операционной системы, с той лишь разницей, что исполняются в рамках браузера, а не в собственном процессе операционной системы.

Для создания Single Page Application (SPA) с разделами "Список владельцев", "Список авто" и "Регистрация авто", создадим отдельную папку для frontend-части. В этой папке будет HTML, CSS и JavaScript файлы.

Файловая структура проекта:

```
mygibdd/  
├─ frontend/  
│   ├─ index.html  
│   ├─ styles.css  
│   └─ script.js  
└─ backend/ (ваша предыдущая backend-часть)
```

1. HTML (файл frontend/index.html)

Шаг 1.

```
1  <!DOCTYPE html>  
2  <html lang="ru">  
3  <head>  
4      <meta charset="UTF-8">  
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">  
6      <title>МояГИБДД</title>  
7      <link rel="stylesheet" href="styles.css">  
8  </head>  
9  <body>  
10     <h1>МояГИБДД</h1>  
11 >     <div class="container">...  
12     </div>  
13  
14  
15     <script src="script.js"></script>  
16 </body>  
17 </html>
```

Шаг 2.

```

11 <div class="container">
12   <!-- Список владельцев -->
13   <section>
14     <h2>Список владельцев</h2>
15     <table id="owners-table">
16       <thead>
17         <tr>
18           <th>ID</th>
19           <th>Фамилия</th>
20           <th>Имя</th>
21           <th>Телефон</th>
22           <th>Почта</th>
23           <th>Город</th>
24         </tr>
25       </thead>
26       <tbody>
27         <!-- Данные будут загружены через JavaScript -->
28       </tbody>
29     </table>
30   </section>
31

```

Шаг 3.

```

32 <!-- Список авто -->
33 <section>
34   <h2>Список авто</h2>
35   <table id="autos-table">
36     <thead>
37       <tr>
38         <th>ID</th>
39         <th>Марка</th>
40         <th>Модель</th>
41         <th>Цвет</th>
42         <th>Госномер</th>
43       </tr>
44     </thead>
45     <tbody>
46       <!-- Данные будут загружены через JavaScript -->
47     </tbody>
48   </table>
49 </section>

```

Шаг 4.

```
51 <!-- Регистрация авто -->
52 <section>
53   <h2>Регистрация авто</h2>
54   <form id="register-auto-form">
55     <label for="owner-select">Владелец:</label>
56     <select id="owner-select">
57       <!-- Опции будут загружены через JavaScript -->
58     </select>
59
60     <label for="auto-select">Автомобиль:</label>
61     <select id="auto-select">
62       <!-- Опции будут загружены через JavaScript -->
63     </select>
64
65     <button type="submit">Зарегистрировать</button>
66   </form>
67
68   <h3>Принадлежащие авто:</h3>
69   <ul id="owned-autos-list">
70     <!-- Данные будут загружены через JavaScript -->
71   </ul>
72 </section>
```

2. CSS (файл frontend/styles.css)

Впишите комментарии для повторения css атрибутов на против строк, значение которых вы не знали или забыли.

Часть 1	Часть 2.
<pre>1 √ body { 2 font-family: Arial, sans-serif; 3 margin: 20px; 4 } 5 6 √ h1, h2 { 7 color: <input type="text" value="#333"/>; 8 } 9 10 √ .container { 11 max-width: 800px; 12 margin: 0 auto; 13 } 14 15 √ table { 16 width: 100%; 17 border-collapse: collapse; 18 margin-bottom: 20px; 19 } 20 21 √ table, th, td { 22 border: 1px solid <input type="text" value="#ddd"/>; 23 } 24 25 √ th, td { 26 padding: 8px; 27 text-align: left; 28 } 29 30 √ th { 31 background-color: <input type="text" value="#f4f4f4"/>; 32 } 33 34 √ form { 35 margin-bottom: 20px; 36 }</pre>	<pre>37 38 label { 39 display: block; 40 margin-bottom: 5px; 41 } 42 43 select, button { 44 padding: 8px; 45 margin-bottom: 10px; 46 width: 100%; 47 max-width: 300px; 48 } 49 50 button { 51 background-color: <input type="text" value="#28a745"/>; 52 color: <input type="text" value="white"/>; 53 border: none; 54 cursor: pointer; 55 } 56 57 button:hover { 58 background-color: <input type="text" value="#218838"/>; 59 } 60 61 ul { 62 list-style-type: none; 63 padding: 0; 64 } 65 66 ul li { 67 background-color: <input type="text" value="#f8f9fa"/>; 68 padding: 8px; 69 margin-bottom: 5px; 70 border: 1px solid <input type="text" value="#ddd"/>; 71 }</pre>

3. JavaScript (файл frontend/script.js)

Шаг 1. Создаем функцию, которая выполняется ПОСЛЕ загрузки HTML DOM-объекта (DOM – это содержимое HTML для открываемой страницы)

```
frontend > JS code_front_local.js > ...
1 > document.addEventListener('DOMContentLoaded', () => { ...
110 });
```

Шаг 2. Создаем асинхронную = работает параллельно с загрузкой страницы HTML DOM – это основной поток. Асинхронность нужна для долгих или рискованных операций, которые задерживают загрузку DOM или работу других функций. Пользователю нельзя показывать медленную страницу.

После выполнения асинхронная функция отправляет сигнал основному потоку, что она закончила работу. Основной поток = отображение страницы может показать то, что подготовила функция loadData()

```
2 // Загружаем данные из JSON (предполагаем, что они доступны по HTTP)
3 async function loadData() {
4   const [users, autos, owners] = await Promise.all([
5     fetch('/data/users.json').then(res => res.json()),
6     fetch('/data/autos.json').then(res => res.json()),
7     fetch('/data/owners.json').then(res => res.json()),
8   ]);
9
10   return { users, autos, owners };
11 }
```

Как это работает:

1. HTML:

- Три секции: "Список владельцев", "Список авто" и "Регистрация авто".
- Таблицы и формы для взаимодействия с данными.

2. CSS:

- Простые стили для оформления таблиц, форм и списков.

3. JavaScript:

- Загружает данные из JSON-файлов (предполагается, что они доступны по HTTP).
- Отображает данные в таблицах и выпадающих списках.
- Позволяет регистрировать авто за владельцами и отображает принадлежащие авто.

Как использовать:

1. Запустите backend-часть (Node.js), чтобы данные были доступны по HTTP.
2. Откройте index.html в браузере.
3. Взаимодействуйте с интерфейсом: просматривайте списки, регистрируйте авто за владельцами.

Теперь у вас есть полноценное SPA для управления данными владельцев и автомобилей!

