

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего
образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
Факультет инфокоммуникационных технологий

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 2
по теме: взаимодействие с внешним API
по дисциплине: Фронт-энд разработка

Специальность:

09.03.03 Мобильные и сетевые технологии

Выполнил:

Рашевский В.В., К33402

Проверил:

Добряков Д.И.

Санкт-Петербург,

2023

Задача.

Варианты остаются прежними. Теперь Вам нужно привязать то, что Вы делали в ЛР1 к внешнему API средствами fetch/axios/xhr.

Вариант.

Разработка интерактивного сайта для управления умным домом: создание пользовательского интерфейса, который позволит пользователям контролировать освещение, температуру и другие параметры в своем доме с помощью веб-приложения.

Решение.

Код программы представлен на Github.

Был выбран fetch и json-server в качестве API.

База данных представлена на рисунке 1.



```
10  },
11  {
12    "id": 2,
13    "title": "CCTV",
14    "text": "You can use CCTV using our website.",
15    "src": "../src/camera.jpg",
16    "alt": "CCTV at front door",
17    "href": "cctv.html"
18  },
19  {
20    "id": 3,
21    "title": "Door locks",
22    "text": "You can close and open your door lock using our website.",
23    "src": "../src/lock.jpg",
24    "alt": "Smart door locks",
25    "href": "door_locks.html"
26  },
27  {
28    "id": 4,
29    "title": "Temperature",
30    "text": "You can control temperature in your house using our website.",
31    "src": "../src/temperature.jpg",
32    "alt": "Temperature controller at wall",
33    "href": "temperature.html"
34  }
35 ],
36 "use_cases": [
37   {
38     "id": 1,
39     "use_case": "Retail"
40   },
41   {
42     "id": 2,
43     "use_case": "Healthcare"
44   },
45   {
46     "id": 3,
47     "use_case": "Smart home"
48   },
49   {
50     "id": 4,
51     "use_case": "Logistics & transportation"
52   },
53   {
54     "id": 5,
55     "use_case": "Agriculture"
56   }
57 ],
58 "about": {
59   "title": "What can we do for you?",
60   "desc": "You can control your area, light, door locks and temperature in house."
61 }
62 }
```

Рисунок 1 – Файл db.json

Без изменения дизайна были заменены встроенные карточки всех сервисов на автоматически генерируемые, что показано на рисунках 2-3, как

и описание компании с ключевыми use cases, что показано на рисунках 4-5, как. Все используемые запросы представлены на рисунке 6.

The image shows a code editor with two files: `home.html` and `api.js`.

home.html contains the following code:

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet"
6     integrity="sha384-T3c6Co1i6uA9TneEoa7b7xnatzjcDSCmG1MXxSR1GAsXEV/DwwyoSc21KP2992jy3v/Wx1te1KdS7gFy1cJ6L93d"
7     crossorigin="anonymous">
8   <link rel="stylesheet" href="/css/main.css">
9   <link rel="stylesheet" href="/css/dark.css" media="(prefers-color-scheme: dark)">
10  <link rel="stylesheet" href="/css/light.css" media="(prefers-color-scheme: light)">
11  <title>All available services</title>
12 </head>
13 <body>
14   <header id="header">
15     <main id="cards" class="container-fluid row">
16       </main>
17     </header>
18     <script src="/js/api.js"></script>
19     <script src="/js/search.js"></script>
20     <script src="/js/auth.js"></script>
21     <script src="/js/auth_script.js"></script>
22     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
23       integrity="sha384-C6RzsynH9K9W0Mh47t87950GyZPChTmX31NW7b8CYNUO81pcV8yq46cDfL"
24       crossorigin="anonymous"></script>
25   </body>
26 </html>

```

api.js contains the following code:

```

55 const mainContainer : HTMLElement = document.getElementById( 'elementId: 'cards' );
56 if (mainContainer) {
57   const list : DocumentFragment = document.createDocumentFragment();
58   fetchServices().then((data) => {
59     console.log(data);
60     data.map(function (service) : void {
61       const div1 : HTMLDivElement = document.createElement( 'div' );
62       div1.className = "col-12 col-xl-4 col-md-6 my-3";
63       const divCard : HTMLDivElement = document.createElement( 'div' );
64       divCard.className = "card";
65       const img : HTMLImageElement = document.createElement( 'img' );
66       img.src = service.src;
67       img.alt = service.alt;
68       img.className = "card-img-top object-fit-contain";
69       img.width = 500;
70       img.height = 500;
71       const divBody : HTMLDivElement = document.createElement( 'div' );
72       divBody.className = "card-body";
73       const h5Title : HTMLHeadingElement = document.createElement( 'h5' );
74       h5Title.className = "card-title";
75       h5Title.innerHTML = `${service.title}`;
76       const pText : HTMLParagraphElement = document.createElement( 'p' );
77       pText.className = "card-text";
78       pText.innerHTML = `${service.text}`;
79       const a : HTMLAnchorElement = document.createElement( 'a' );
80       a.className = "btn btn-primary";
81       a.href = `${service.href}`;
82       a.innerHTML = `Check ${service.title} service`;
83       divBody.appendChild(h5Title);
84       divBody.appendChild(pText);
85       divBody.appendChild(a);
86       divCard.appendChild(img);
87       divCard.appendChild(divBody);
88       div1.appendChild(divCard);
89       console.log(div1);
90       list.appendChild(div1);
91     });
92   }).finally(() => {
93     mainContainer.appendChild(list);
94   });
95 }

```

Рисунок 2 – Код для страницы сервисов

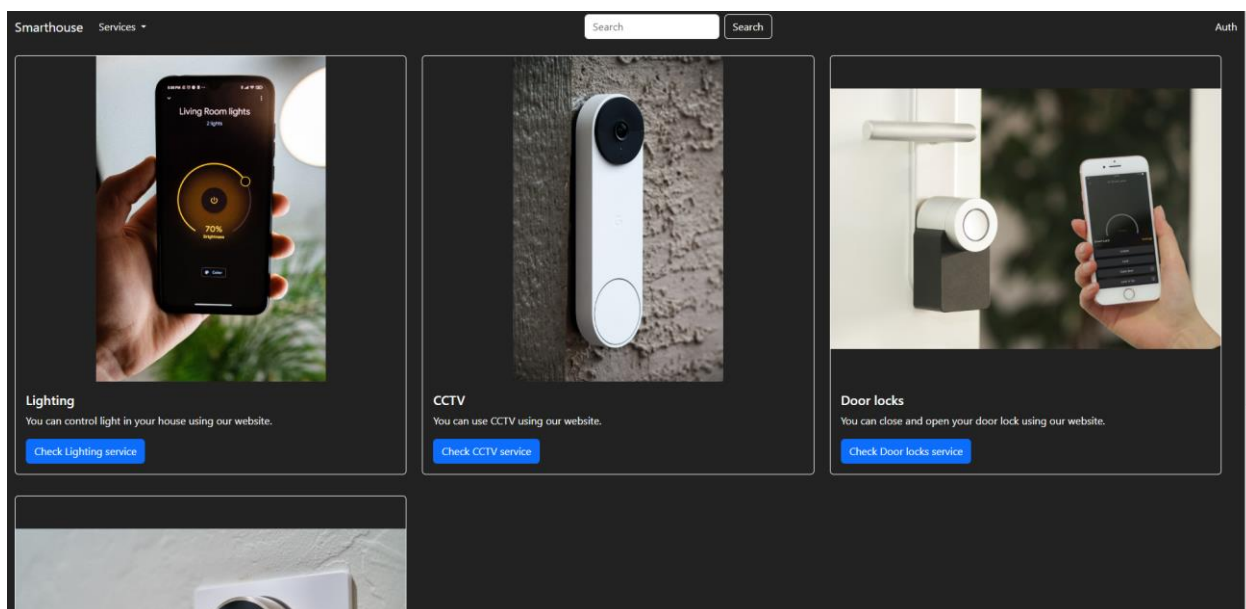


Рисунок 3 – Страница «All services»

```

44         aria-label="Search" id="searchInput">
45         <datalist id="datalistOptions">
46         </datalist>
47         <input class="btn btn-outline-success" type="submit" value="Search">
48
49     </form>
50     <div class="d-flex mx-auto">
51     <a id="account" class="nav-link active align-content-end ms-3" href="#account.html">Account</a>
52     <a id="auth" class="nav-link active align-content-end ms-3" href="#auth.html">Auth</a>
53     <a id="exit" class="nav-link active align-content-end ms-3" href="#home.html"
54         onClick="exit()">Exit</a>
55     </div>
56 </div>
57 </nav>
58 </header>
59 <main class="container-fluid align-content-center row vh-100">
60     <div class="justify-content-center row">
61     <div class="card col-12 col-xl-5 col-md-6 my-5 mx-auto">
62     <h5 class="card-header">Key use cases</h5>
63     <div class="card-body">
64     <ul class="list-group" id="useCasesList">
65     </ul>
66     </div>
67 </div>
68 <div class="container card col-12 col-xl-5 col-md-6 my-5 mx-auto">
69     <h5 class="card-header">What you can achieve with our services</h5>
70     <div class="card-body">
71     <div class="list-group">
72     <div class="list-item">
73     <div class="list-item-content">
74     <div class="list-item-title">
75     <div class="list-item-description">
76     </div>
77     </div>
78     </div>
79     </div>
80     </div>
81     </div>
82     </div>
83     </div>
84     </div>
85     </div>
86     </div>
87     </div>
88     </div>
89     </div>
90     </div>
91     </div>
92     </div>
93     </div>
94     </div>
95     </div>
96     </div>
97     </div>
98     </div>
99     </div>
100    </div>
101    </div>
102    </div>
103    </div>
104    </div>
105    </div>
106    </div>
107    </div>
108    </div>
109    </div>
110    </div>
111    </div>
112    </div>
113    </div>
114    </div>
115    </div>
116    </div>
117    </div>
118    </div>
119    </div>
120    </div>
121    </div>
122    </div>
123    </div>
124    </div>
125    </div>
126    </div>
127    </div>
128    </div>
129    </div>
130    </div>
131    </div>
132    </div>
133    </div>
134    </div>
135    </div>
136    </div>
137    </div>
138    </div>
139    </div>
140    </div>
141    </div>
142    </div>
143    </div>
144    </div>
145    </div>
146    </div>
147    </div>
148    </div>
149    </div>
150    </div>
151    </div>
152    </div>
153    </div>
154    </div>
155    </div>
156    </div>
157    </div>
158    </div>
159    </div>
160    </div>
161    </div>
162    </div>
163    </div>
164    </div>
165    </div>
166    </div>
167    </div>
168    </div>
169    </div>
170    </div>
171    </div>
172    </div>
173    </div>
174    </div>
175    </div>
176    </div>
177    </div>
178    </div>
179    </div>
180    </div>
181    </div>
182    </div>
183    </div>
184    </div>
185    </div>
186    </div>
187    </div>
188    </div>
189    </div>
190    </div>
191    </div>
192    </div>
193    </div>
194    </div>
195    </div>
196    </div>
197    </div>
198    </div>
199    </div>
200    </div>
201    </div>
202    </div>
203    </div>
204    </div>
205    </div>
206    </div>
207    </div>
208    </div>
209    </div>
210    </div>
211    </div>
212    </div>
213    </div>
214    </div>
215    </div>
216    </div>
217    </div>
218    </div>
219    </div>
220    </div>
221    </div>
222    </div>
223    </div>
224    </div>
225    </div>
226    </div>
227    </div>
228    </div>
229    </div>
230    </div>
231    </div>
232    </div>
233    </div>
234    </div>
235    </div>
236    </div>
237    </div>
238    </div>
239    </div>
240    </div>
241    </div>
242    </div>
243    </div>
244    </div>
245    </div>
246    </div>
247    </div>
248    </div>
249    </div>
250    </div>
251    </div>
252    </div>
253    </div>
254    </div>
255    </div>
256    </div>
257    </div>
258    </div>
259    </div>
260    </div>
261    </div>
262    </div>
263    </div>
264    </div>
265    </div>
266    </div>
267    </div>
268    </div>
269    </div>
270    </div>
271    </div>
272    </div>
273    </div>
274    </div>
275    </div>
276    </div>
277    </div>
278    </div>
279    </div>
280    </div>
281    </div>
282    </div>
283    </div>
284    </div>
285    </div>
286    </div>
287    </div>
288    </div>
289    </div>
290    </div>
291    </div>
292    </div>
293    </div>
294    </div>
295    </div>
296    </div>
297    </div>
298    </div>
299    </div>
300    </div>
301    </div>
302    </div>
303    </div>
304    </div>
305    </div>
306    </div>
307    </div>
308    </div>
309    </div>
310    </div>
311    </div>
312    </div>
313    </div>
314    </div>
315    </div>
316    </div>
317    </div>
318    </div>
319    </div>
320    </div>
321    </div>
322    </div>
323    </div>
324    </div>
325    </div>
326    </div>
327    </div>
328    </div>
329    </div>
330    </div>
331    </div>
332    </div>
333    </div>
334    </div>
335    </div>
336    </div>
337    </div>
338    </div>
339    </div>
340    </div>
341    </div>
342    </div>
343    </div>
344    </div>
345    </div>
346    </div>
347    </div>
348    </div>
349    </div>
350    </div>
351    </div>
352    </div>
353    </div>
354    </div>
355    </div>
356    </div>
357    </div>
358    </div>
359    </div>
360    </div>
361    </div>
362    </div>
363    </div>
364    </div>
365    </div>
366    </div>
367    </div>
368    </div>
369    </div>
370    </div>
371    </div>
372    </div>
373    </div>
374    </div>
375    </div>
376    </div>
377    </div>
378    </div>
379    </div>
380    </div>
381    </div>
382    </div>
383    </div>
384    </div>
385    </div>
386    </div>
387    </div>
388    </div>
389    </div>
390    </div>
391    </div>
392    </div>
393    </div>
394    </div>
395    </div>
396    </div>
397    </div>
398    </div
```

Рисунок 4 – Код для главной страницы

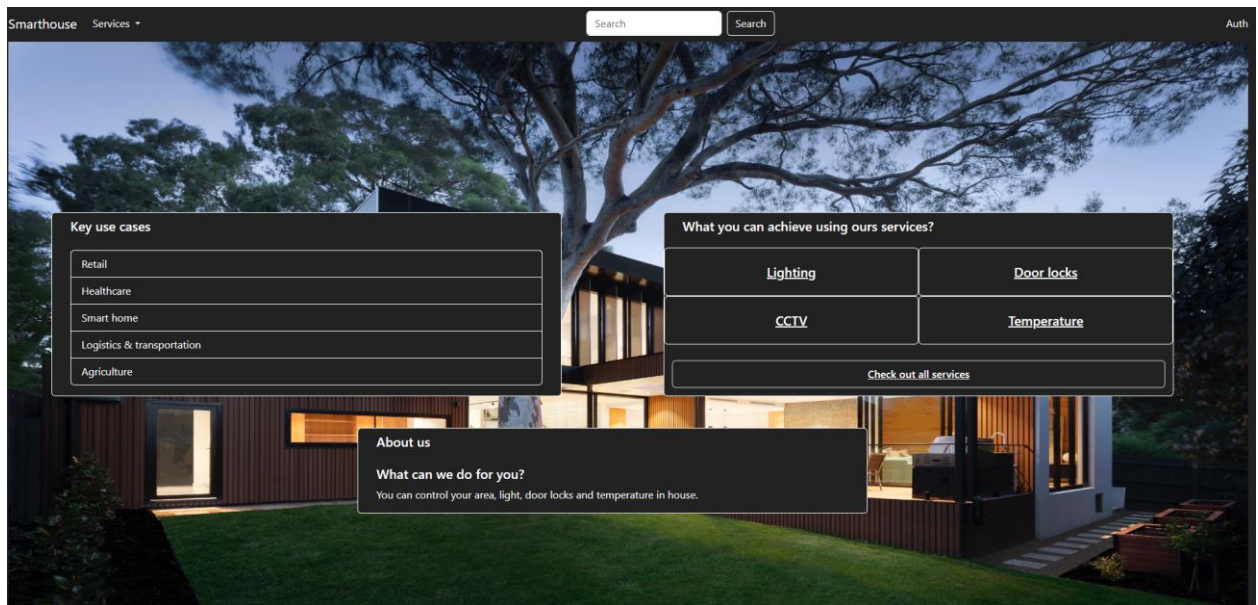


Рисунок 5 – Главная страница

```

1 usage  ▲ rashevskii.vv
2
3 1 async function fetchUseCases() : Promise<any> {
4     const response : Response = await fetch( input: "http://localhost:3000/use_cases");
5     if (!response.ok) {
6         const msg : string = `An error has occurred: ${response.status}, ${response.statusText}`;
7         throw new Error(msg);
8     }
9     return await response.json();
10 }
11
12 1 usage  ▲ rashevskii.vv
13
14 9 async function fetchServices() : Promise<any> {
15     const response : Response = await fetch( input: "http://localhost:3000/services");
16     if (!response.ok) {
17         const msg : string = `An error has occurred: ${response.status}, ${response.statusText}`;
18         throw new Error(msg);
19     }
20     return await response.json();
21 }
22
23 1 usage  ▲ rashevskii.vv
24
25 17 async function fetchAbout() : Promise<any> {
26     const response : Response = await fetch( input: "http://localhost:3000/about");
27     if (!response.ok) {
28         const msg : string = `An error has occurred: ${response.status}, ${response.statusText}`;
29         throw new Error(msg);
30     }
31     return await response.json();
32 }
33
34

```

Рисунок 6 – Код для запросов

Выводы.

Запрограммированы необходимые запросы, создан json-server и обновлены HTML-страницы.