

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИТМО**

Факультет ИКТ

ЛАБОРАТОРНАЯ РАБОТА №3
по дисциплине: «Frontend-разработка»
на тему:
«Разработка одностраничного веб-приложения (SPA) с использованием
фреймворка Vue.JS»

Выполнила:
студент группы К33402
Афанасьева Ирина

Преподаватель: Добряков Давид Ильич

Санкт-Петербург
2022

Цель работы: Мигрировать ранее написанный сайт на фреймворк Vue.JS.

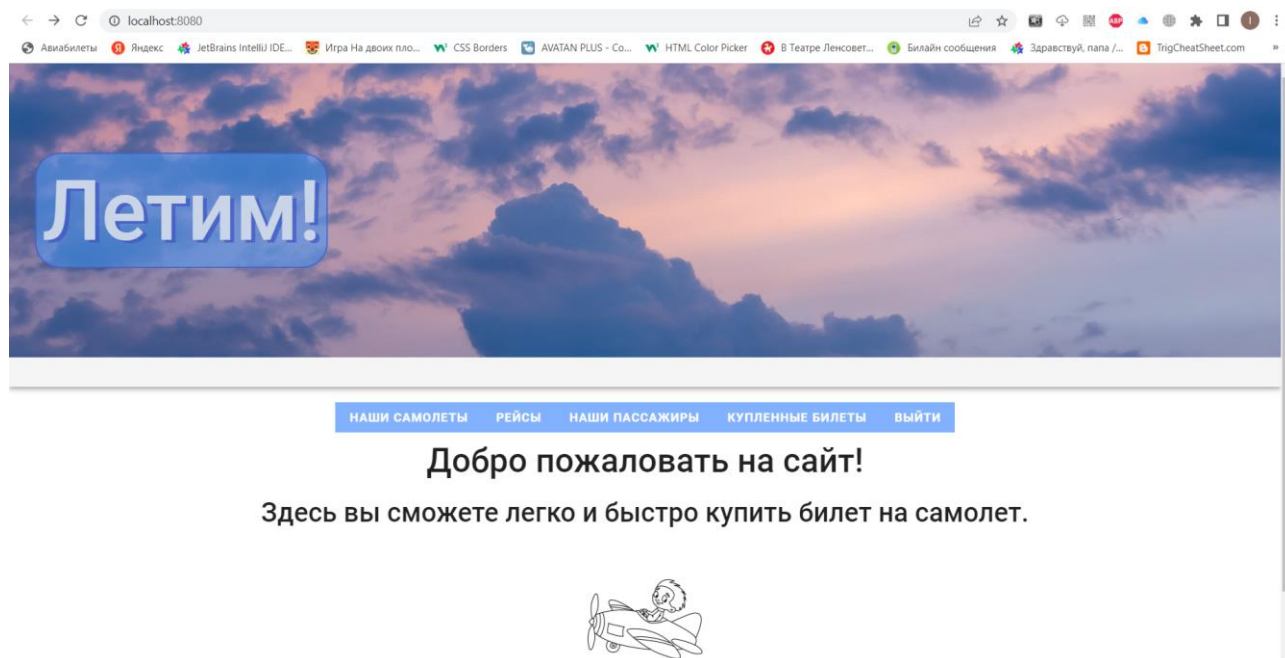
Минимальные требования:

- Должен быть подключён роутер
- Должна быть реализована работа с внешним API
- Разумное деление на компоненты

Вариант: Сайт для покупки билетов на воздушные рейсы

Ход работы:

Реализована клиентская сторона для сервиса покупки авиабилетов.



Рутинг(index.js).

```
1 import Vue from 'vue'
2 import VueRouter from 'vue-router'
3 import Greeting from '../views/Greeting.vue'
4 import Log from '../views/Log.vue'
5 import Logout from '../views/Logout.vue'
6 import Reg from '../views/Reg.vue'
7 import Planes from '../views/Planes.vue'
8 import Flight from '../views/Flight.vue'
9 import Tickets from '../views/Tickets.vue'
10 import Passengers from '../views/Passengers.vue'
11 import FlightCreate from '../views/FlightCreate.vue'
12 import FlightUpd from '../views/FlightUpd.vue'
13 import CreateTicket from '../views/CreateTicket.vue'
```

```

15     Vue.use(VueRouter)
16
17     const routes = [
18       {
19         path: '/',
20         name: 'Greeting',
21         component: Greeting
22       },
23       {
24         path: '/log',
25         name: 'Log',
26         component: Log
27       },
28       {
29         path: '/logout',
30         name: 'Logout',
31         component: Logout
32       },
33       {
34         path: '/reg',
35         name: 'Reg',
36         component: Reg
37       },
38       {
39         path: '/planes',
40         name: 'Planes',
41         component: Planes
42       },

```

```

43     {
44       path: '/tickets',
45       name: 'Tickets',
46       component: Tickets
47     },
48     {
49       path: '/flight/:flight_id/ticket',
50       name: 'CreateTicket',
51       component: CreateTicket
52     },

```

```

54     path: '/flight',
55     name: 'Flight',
56     component: Flight
57   },
58   {
59     path: '/flight/create',
60     name: 'FlightCreate',
61     component: FlightCreate
62   },
63   {
64     path: '/flight/upd/:flight_id/',
65     name: 'FlightUpd',
66     component: FlightUpd
67   },
68   {
69     path: '/passengers',
70     name: 'Passengers',
71     component: Passengers
72   },
73   {
74     path: '/flights/create',
75     name: 'FlightCreate',
76     component: FlightCreate
77   },

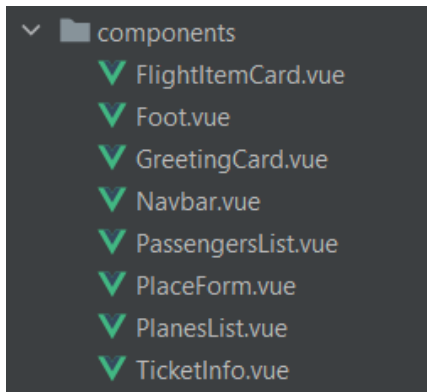
```

```

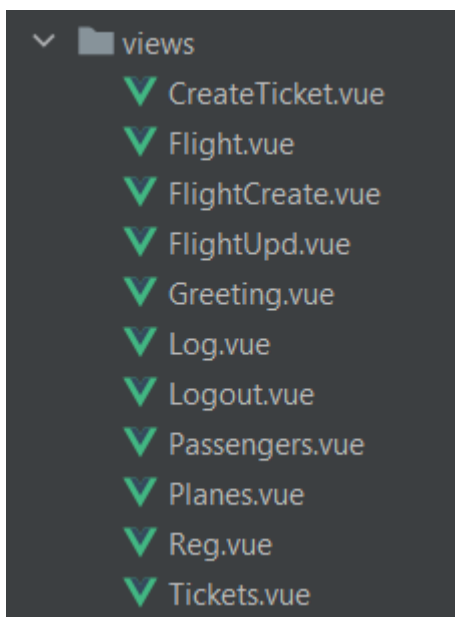
80   const router = new VueRouter( options: {
81     mode: 'history',
82     base: process.env.BASE_URL,
83     routes
84   })
85
86   2 usages  Username1-beep
87   export default router

```

Components:



Views:



App.vue

```
1  <template>
2    <v-app>
3      <v-app-bar :src="require('./assets/planehed.jpg')" height="350px;" contain>
4        <div class="headerPic">
5          <h1>Летим!</h1>
6        </div>
7      </v-app-bar>
8
9      <v-main class="d-flex align-center text-center">
10       <v-content style="height: 300px">
11         <navbar />
12         <router-view />
13         <foot />
14       </v-content>
15
16     </v-main>
17   </v-app>
18 </template>
19
20 <script>
21 import ...
22
23
24 export default {
25   name: "App",
26   components: { Navbar, Foot },
27   data: () => ({
28     //
29   }),
30 };
31 </script>
```

```
32 <style>
33   .headerPic {
34     width: 100%;
35     margin: 0;
36     display: flex;
37     background-size: cover;
38     justify-content: flex-start;
39     align-items: flex-end;
40   }
41   .headerPic h1 {
42     border: 2px solid #3f57c4bb;
43     border-radius: 30px;
44     margin: 15px;
45     padding: 7px;
46     background-color: rgba(80, 149, 239, 0.6);
47     color: rgb(202, 213, 231);
48     text-shadow: 5px 3px #3f57c4bb;
49     font-size: 100px;
50   }
51 </style>
```

Список самолётов

Страница содержит данные о всех самолётах, значащихся в системе аэропорта.

URL : `/planes`

Methods : `GET`

Success Code : `200 OK`

Код:

```
1 <template>
2   <section>
3     <h1 style="...">Список наших самолетов</h1>
4     <v-row>
5       <v-col cols="5" class="mx-auto">
6         <v-card
7           elevation="2"
8           class="px-10 py-5"
9           color="white"
10          >
11           <planes-list :planes="planes" />
12         </v-card>
13         <br>
14         <v-btn
15           color="accent"
16           class="mb-3 mr-3"
17           @click="$router.push(`/`)"
18           >На главную</v-btn>
19       </v-col>
20     </v-row>
21   </section>
22 </template>
23
24 <script>
25 import ...
26
27 export default {
28   name: 'Planes',
```

```

33     components: {
34         PlanesList,
35     },
36
37
38     data: () => ({
39         planes: [],
40
41     }),
42
43
44     methods: {
45         async PlanesList () {
46             try {
47                 const response = await axios
48                     .get( url: 'http://127.0.0.1:8000/planes/all')
49                 if (response.status !== 200) {
50                     throw new Error(response.status)
51                 }
52                 this.planes = response.data
53                 return response.data
54             } catch (e) {
55                 console.error('AN API ERROR', e)
56             }
57         },
58     },
59

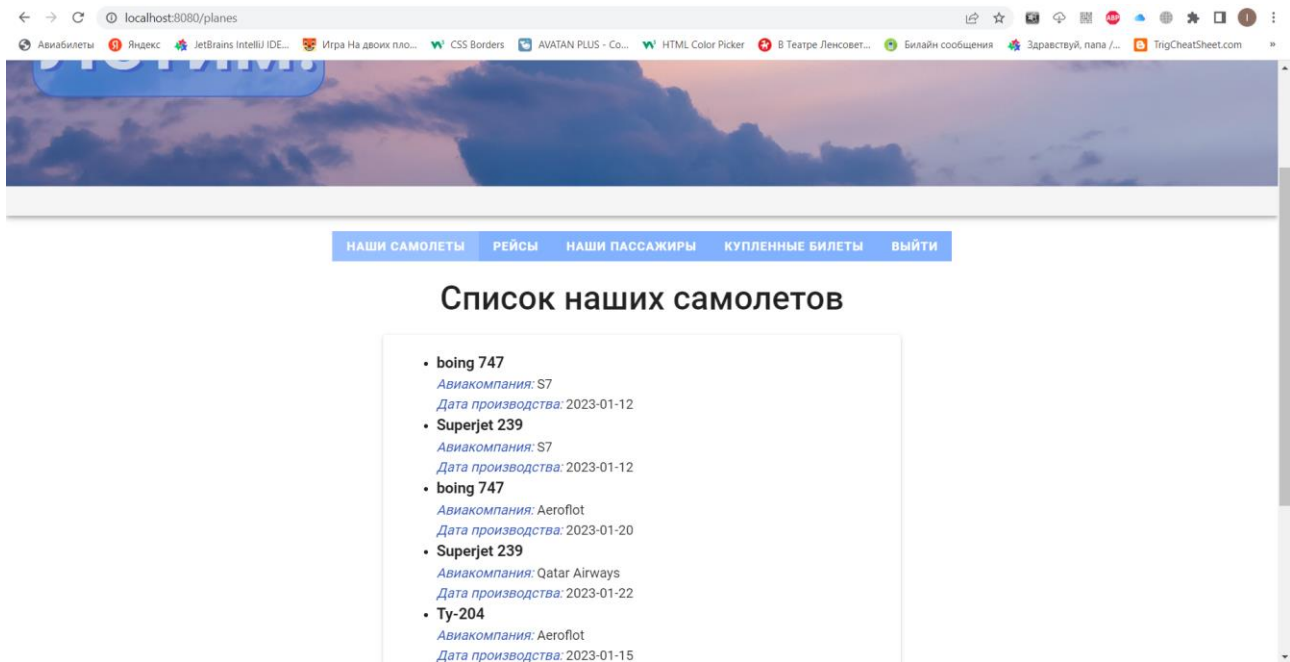
```

```

60     created () {
61         this.PlanesList()
62     }
63 }
64 </script>
65
66 <style>
67
68 </style>

```


Вид:



localhost:8080/planes

Авиабилеты Яндекс JetBrains IntelliJ IDE... Игра На двоих пло... CSS Borders AVATAN PLUS - Co... HTML Color Picker В Театре Ленсовет... Биллайн сообщения Зараствуй, нана /... TrigCheatSheet.com

НАШИ САМОЛЕТЫ РЕЙСЫ НАШИ ПАССАЖИРЫ КУПЛЕННЫЕ БИЛЕТЫ ВЫЙТИ

Список наших самолетов

- **boing 747**
Авиакомпания: S7
Дата производства: 2023-01-12
- **Superjet 239**
Авиакомпания: S7
Дата производства: 2023-01-12
- **boing 747**
Авиакомпания: Aeroflot
Дата производства: 2023-01-20
- **Superjet 239**
Авиакомпания: Qatar Airways
Дата производства: 2023-01-22
- **Ту-204**
Авиакомпания: Aeroflot
Дата производства: 2023-01-15

Список Рейсов

Страница содержит данные о рейсах с возможность фильтрации по пункту отправления. Доступно изменение и удаление существующих записей, а также добавление новых.

URL : `/flight`

Methods : `GET` , `POST` , `PUT` , `DELETE`

Success Code : `200 OK`

Код:

```
1 <template>
2 <section>
3   <place-form :from="savedPlace" />
4   <v-row>
5     <v-form @submit.prevent="submitPlace" ref="form" class="my-2">
6       <v-row>
7         <v-col cols="3" class="mx-auto">
8           <v-text-field
9             label="Введите пункт вылета"
10            v-model="from"
11            name="from"
12            placeholder="Spb"
13          />
14        </v-col>
15      </v-row>
16    </v-form>
17    <v-col cols="6" class="mx-auto">
18      <flight-item-card
19        v-for="flightItem in flightItems"
20        :key="flightItem.identifier"
21        :flight-item="flightItem"
22        class="my-2"
23      />
24      <v-btn
25        color="#3f57c4bb"
26        class="mb-3 mr-3"
27        @click="$router.push('/flight/create/')"
28      >Добавить рейс</v-btn>
29    >
30    <br />
```

```

31     <v-btn color="#3f57c4bb" class="mb-3 mr-3" @click="$router.push('/')"  

32     >На главную</v-btn  

33     >  

34   </v-col>  

35 </v-row>  

36 </section>  

37 </template>

```

```

38  

39 <script>  

40 import ...  

41  

42  

43 export default {  

44   components: { FlightItemCard, PlaceForm },  

45   name: "Flight",  

46  

47   data: () => ({  

48     flightItems: [],  

49     from: "",  

50   })),  

51  

52   methods: {  

53     submitPlace() {  

54       localStorage.setItem("from", this.from);  

55       this.savedPlace = this.from;  

56  

57       this.$refs.form.reset();  

58       this.getFlightItems();

```

```

60     async getFlightItems() {  

61       try {  

62         const response = await this.axios.get(  

63           url: `http://127.0.0.1:8000/flights/all?from=${this.savedPlace}`  

64         );  

65  

66         if (response.status !== 200) {  

67           throw new Error(response.error);  

68         }  

69  

70         const flightItems = response.data.map((flightItem) => {  

71           let date = new Date(flightItem.departure);  

72  

73           let year = date.getFullYear();  

74  

75           let month =  

76             String( value: date.getMonth() + 1).length > 1  

77               ? date.getMonth() + 1  

78               : `0${date.getMonth() + 1}`;

```

```

80     let day =
81         String(date.getDate()).length > 1
82         ? date.getDate()
83         : `0${date.getDate()}`;
84     let hour =
85         String(date.getHours()).length > 1
86         ? date.getHours()
87         : `0${date.getHours()}`;
88     let min =
89         String(date.getMinutes()).length > 1
90         ? date.getMinutes()
91         : `0${date.getMinutes()}`;
92
93     flightItem.departure = `${year}/${month}/${day} ${hour}:${min}`;
94
95     date = new Date(flightItem.arrival);
96
97     year = date.getFullYear();
98
99     month =
100         String( value: date.getMonth() + 1).length > 1
101         ? date.getMonth() + 1
102         : `0${date.getMonth() + 1}`;
103
104     day =
105         String(date.getDate()).length > 1
106         ? date.getDate()
107         : `0${date.getDate()}`;

```

```

108     hour =
109         String(date.getHours()).length > 1
110         ? date.getHours()
111         : `0${date.getHours()}`;
112     min =
113         String(date.getMinutes()).length > 1
114         ? date.getMinutes()
115         : `0${date.getMinutes()}`;
116
117     flightItem.arrival = `${year}/${month}/${day} ${hour}:${min}`;
118
119     return flightItem;
120 });

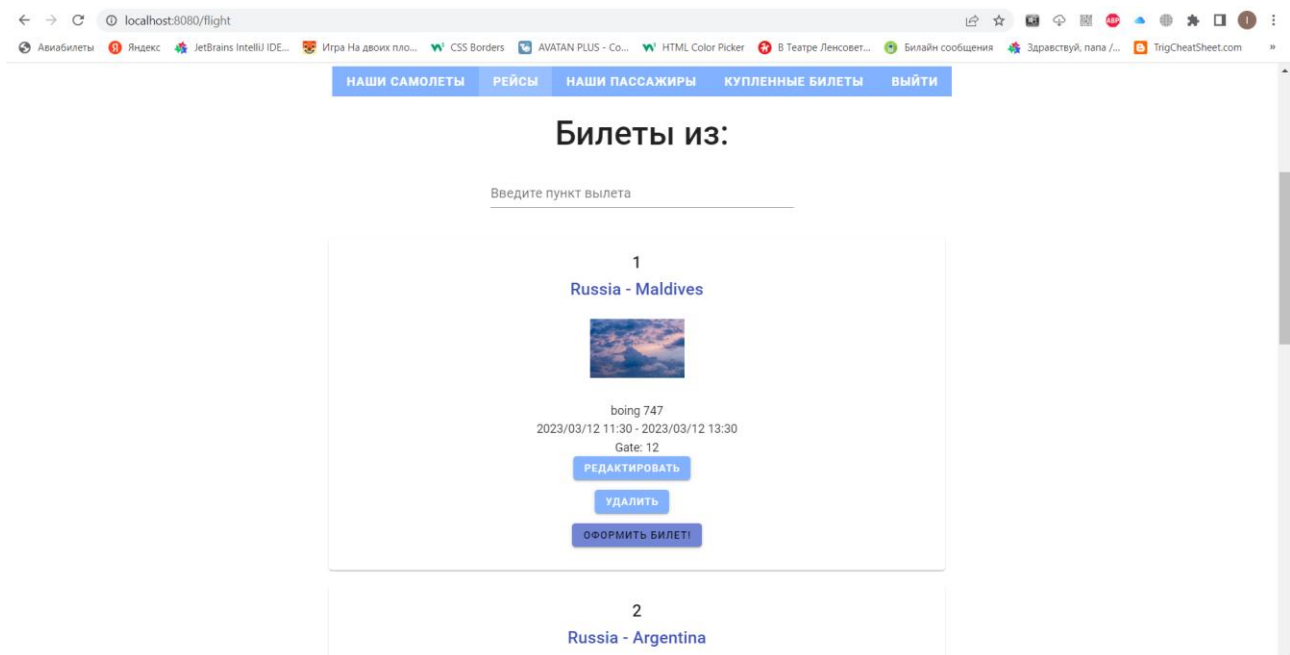
```

```

122         this.flightItems = flightItems;
123     } catch (e) {
124         console.error("AN API ERROR", e);
125     }
126 },
127 },
128
129     created() {
130         if (localStorage.getItem( key: "from")) this.from = localStorage.getItem( key: "from");
131         this.savedPlace = this.from;
132         this.getFlightItems();
133     },
134 };
135 </script>
136
137 <style>
138 </style>

```

Вид:



Список пассажиров:

Страница содержит данные о зарегистрированных пассажирах.

URL : `/passengers`

Methods : `GET`

Success Code : `200 OK`

Код:

```
1 <template>
2   <section>
3     <h1 style="...">Наши пассажиры</h1>
4     <v-row>
5       <v-col cols="4" class="mx-auto">
6         <v-card
7           elevation="2"
8           class="px-10 py-5"
9           color= "#3f57c4bb"
10          dark
11         >
12           <passengers-list :passengers="passengers" />
13         </v-card>
14         <br>
15         <v-btn
16           color="accent"
17           class="mb-3 mr-3"
18           @click="$router.push(`/`)"
19           >На главную</v-btn>
20         </v-col>
21       </v-row>
22     </v-row>
23   </section>
24 </template>
25
```

```

27 <script>
28 import ...
29
30
31 export default {
32   name: 'Passengers',
33
34   components: {
35     PassengersList,
36   },
37
38
39   data: () => ({
40     passengers: [],
41
42   }),
43
44
45   methods: {
46     async PassengersList () {
47       try {
48         const response = await axios
49           .get( url: 'http://127.0.0.1:8000/passengers/all')
50         if (response.status !== 200) {
51           throw new Error(response.status)
52         }
53         this.passengers = response.data
54         return response.data
55       } catch (e) {
56         console.error('AN API ERROR', e)
57       }

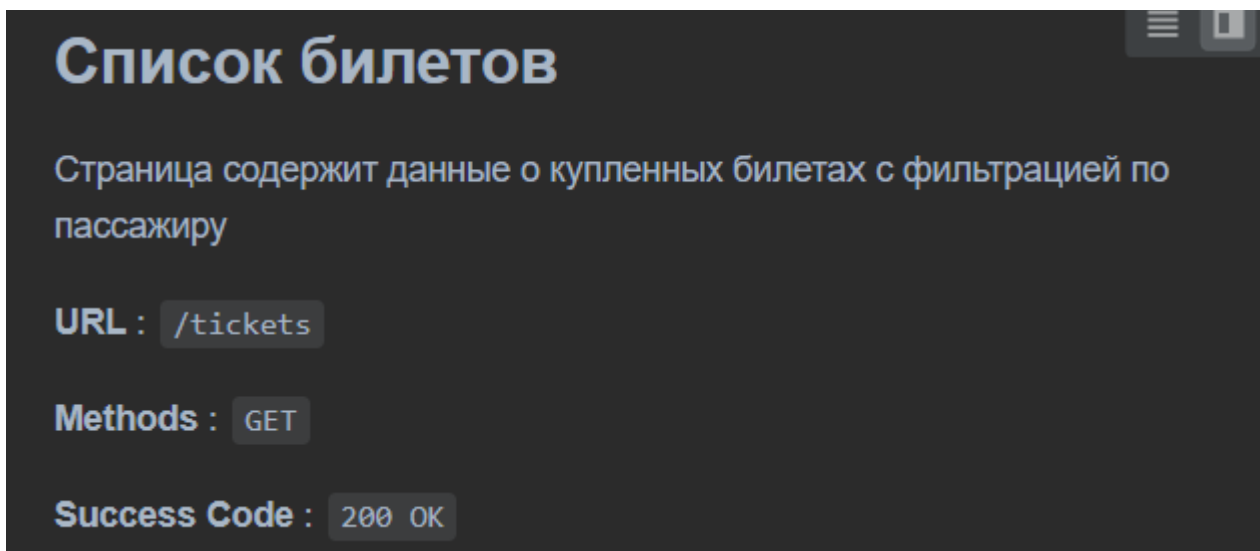
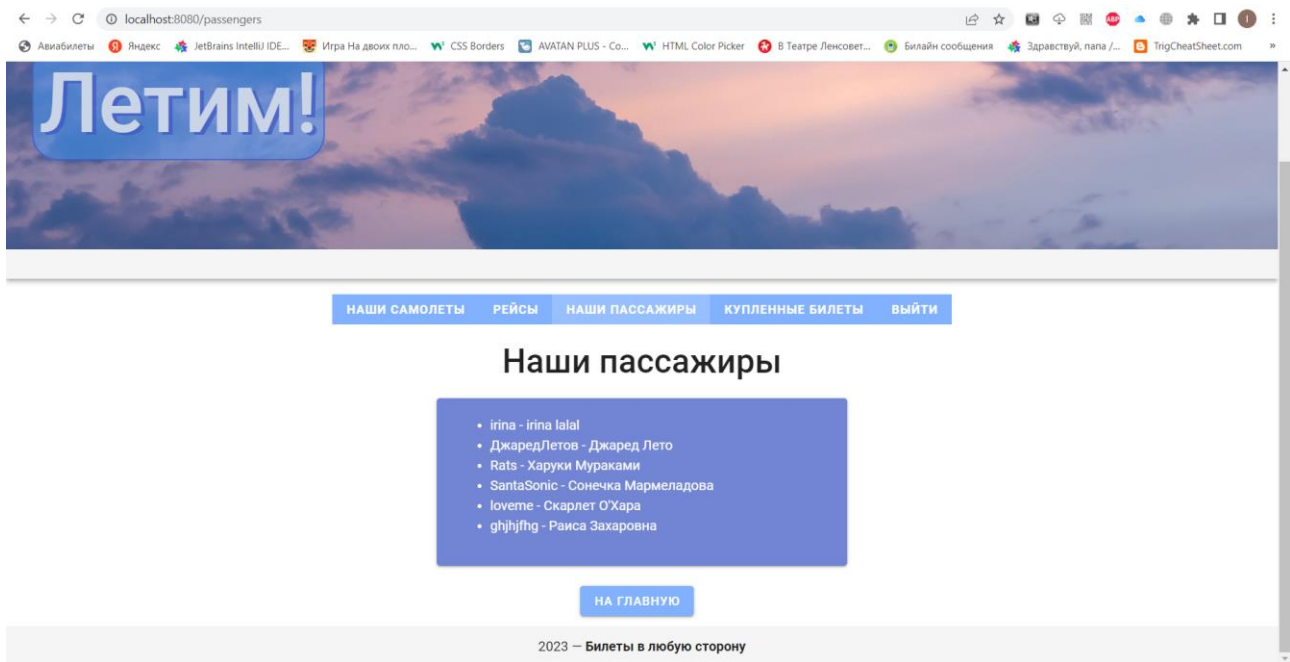
```

```

58     },
59   },
60
61   created () {
62     this.PassengersList()
63   }
64 }
65 </script>
66
67 <style>
68
69 </style>

```

Вид:



Код:

```
1 <template>
2   <section>
3     <h1 style="...">Билеты</h1>
4     <v-row>
5       <v-form @submit.prevent="submitPas" ref="form" class="my-2">
6         <v-row>
7           <v-col cols="3" class="mx-auto">
8             <v-select
9               label="Пассажир"
10              v-model="passenger"
11              :items="passengers"
12              item-text="username"
13              item-value="id"
14            >
15          </v-select>
16          <br />
17          <v-btn color="accent" class="mb-3 mr-3" @click="$router.push('/')">На главную</v-btn>
18        >
19      </v-col>
20    </v-row>
21  </v-form>
22  <v-col cols="5" class="mx-auto">
23    <ticket-info
24      v-for="ticket in tickets"
25      :key="ticket.identifier"
26      :ticket="ticket"
27      class="my-2"
28    />
29  </v-col>
30 </v-row>
31 </section>
32 </template>
```

```
31 </v-row>
32 </section>
33 </template>
34
35 <script>
36 import ...
37
38
39 export default {
40   components: { TicketInfo },
41   name: "Tickets",
42
43   data: () => ({
44     tickets: [],
45     passengers: [],
46     passenger: "",
47   }),
```

```

49 methods: {
50   async PassengersList() {
51     try {
52       const response = await axios.get(
53         url: "http://127.0.0.1:8000/passengers/all"
54       );
55       if (response.status !== 200) {
56         throw new Error(response.status);
57       }
58       this.passengers = response.data;
59
60       return response.data;
61     } catch (e) {
62       console.error("AN API ERROR", e);
63     }
64   },
65   submitPas() {
66     localStorage.setItem("passenger", this.passenger);
67     this.savedPas = this.passenger;
68     this.$refs.form.reset();
69     this.gettickets();
70   },
71   async gettickets() {
72     try {
73       const response = await this.axios.get(
74         url: `http://127.0.0.1:8000/tickets/all?passenger=${this.savedPas}`
75       );

```

```

77       if (response.status !== 200) {
78         throw new Error(response.error);
79       }
80
81       const tickets = response.data.map((ticket) => {
82         let date = new Date(ticket.flight.departure);
83
84         // собираем дату в нужном виде
85         let year = date.getFullYear();
86

```

```

87     let month =
88         String( value: date.getMonth() + 1).length > 1
89         ? date.getMonth() + 1
90         : `0${date.getMonth() + 1}`;
91
92     let day =
93         String(date.getDate()).length > 1
94         ? date.getDate()
95         : `0${date.getDate()}`;
96     let hour =
97         String(date.getHours()).length > 1
98         ? date.getHours()
99         : `0${date.getHours()}`;
100    let min =
101        String(date.getMinutes()).length > 1
102        ? date.getMinutes()
103        : `0${date.getMinutes()}`;
104
105    ticket.flight.departure = `${year}/${month}/${day} ${hour}:${min}`;
106
107    date = new Date(ticket.flight.arrival);
108
109    year = date.getFullYear();
110
111    month =
112        String( value: date.getMonth() + 1).length > 1
113        ? date.getMonth() + 1
114        : `0${date.getMonth() + 1}`;

```

```

116        day =
117            String(date.getDate()).length > 1
118            ? date.getDate()
119            : `0${date.getDate()}`;
120        hour =
121            String(date.getHours()).length > 1
122            ? date.getHours()
123            : `0${date.getHours()}`;
124        min =
125            String(date.getMinutes()).length > 1
126            ? date.getMinutes()
127            : `0${date.getMinutes()}`;

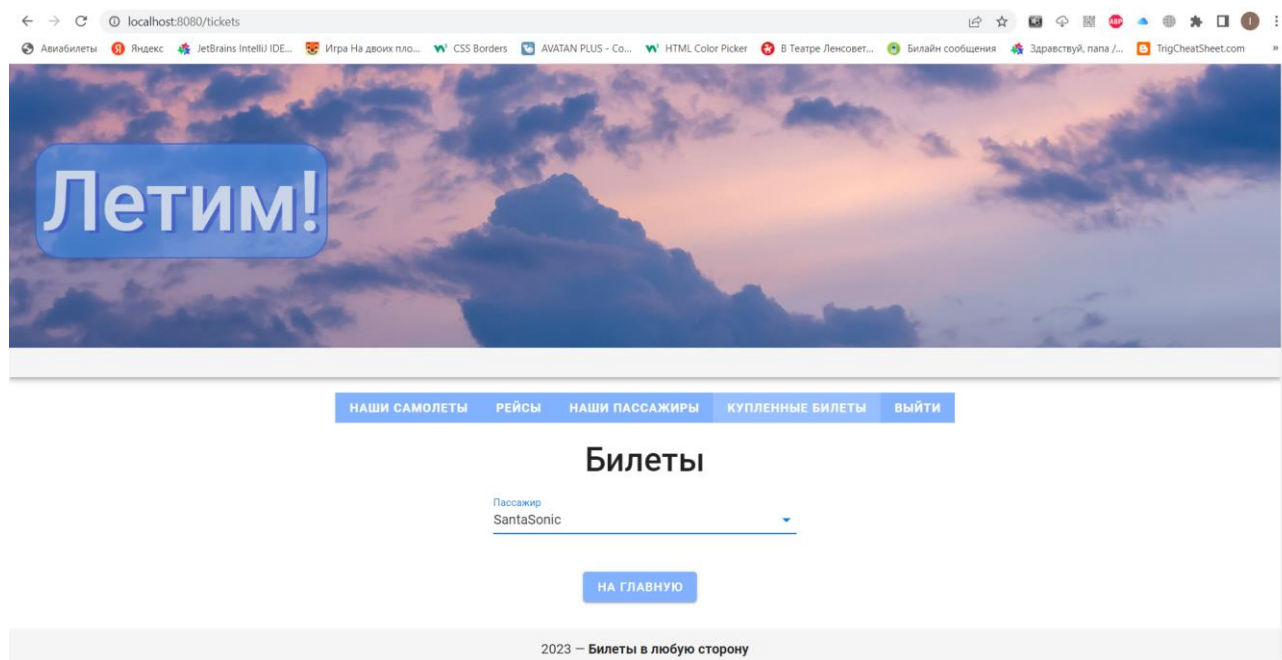
```

```

129         ticket.flight.arrival = `${year}/${month}/${day} ${hour}:${min}`;
130
131         return ticket;
132     });
133
134     this.tickets = tickets;
135 } catch (e) {
136     console.error("AN API ERROR", e);
137 }
138 },
139 },
140
141 created() {
142     this.PassengersList();
143     if (localStorage.getItem( key: "passenger"))
144         this.passenger = localStorage.getItem( key: "passenger");
145     this.savedPlace = this.passenger;
146     this.gettickets();
147 },
148 };
149 </script>
150
151 <style>
152

```

Вид:



Вывод: в ходе этой работы я познакомилась с vue.js, узнала на практике, что такое компонентный подход и реализовала клиентскую часть для своего сервиса.