

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 № 3**  
по теме: Разработка одностраничного веб-приложения (SPA)  
с использованием фреймворка Vue.JS  
по дисциплине: Фронт-энд разработка

Специальность:

09.03.03 Мобильные и сетевые технологии

Выполнил:

Малютина Е.А., К33402

Проверил:

Добряков Д.И.

Санкт-Петербург,

2023

## Задача.

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.
- Использование composable.

## Вариант.

Создание блога, где можно публиковать статьи на различные темы.

Можно добавить функции комментирования, просмотра статистики и подписки на обновления.

## Решение.

Код программы представлен на Github.

Были внесены незначительные корректировки в дизайн. База данных не была изменена, а поиск был исправлен. Исходный код ключевых логических операций и сами страницы представлены на рисунках 1-9.

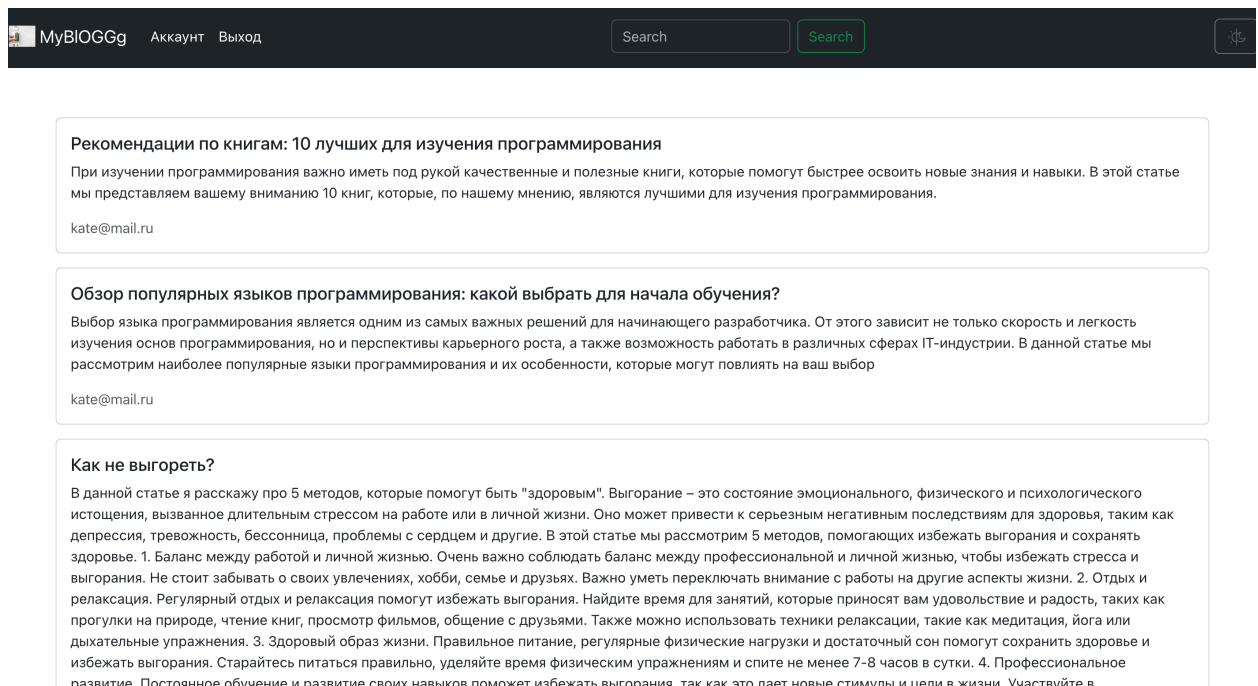


Рисунок 1 – Главная страница в светлой теме

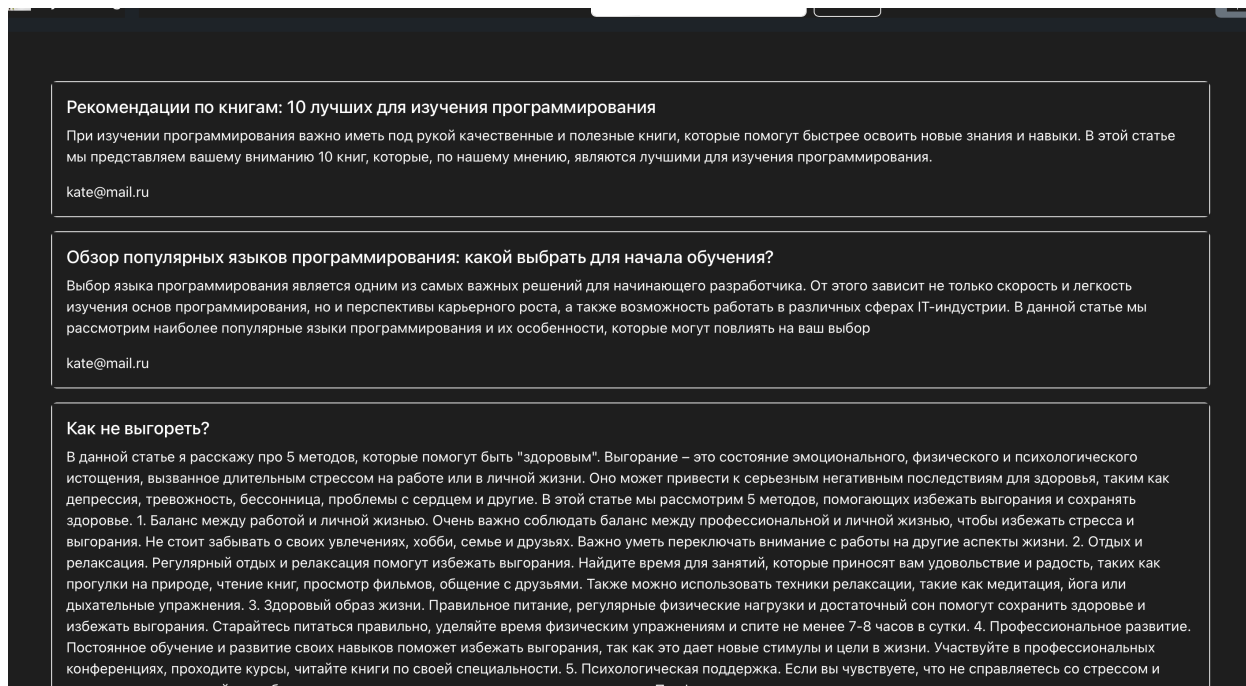


Рисунок 2 – Главная страница в темной теме

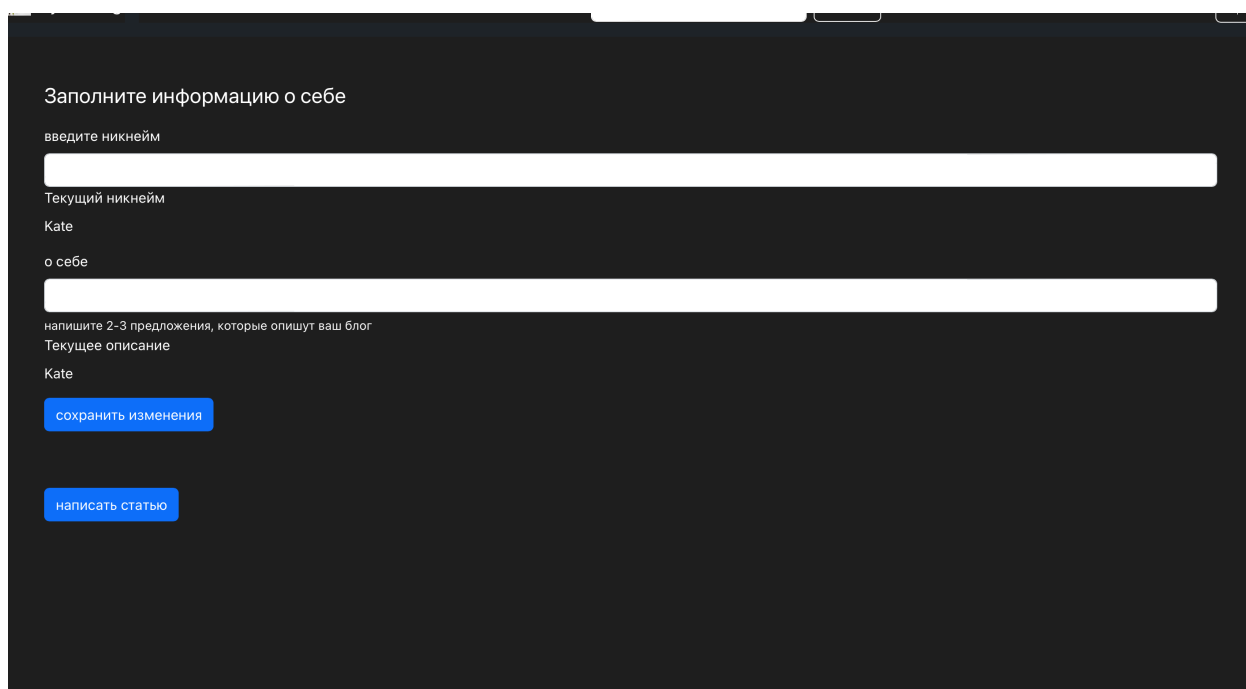
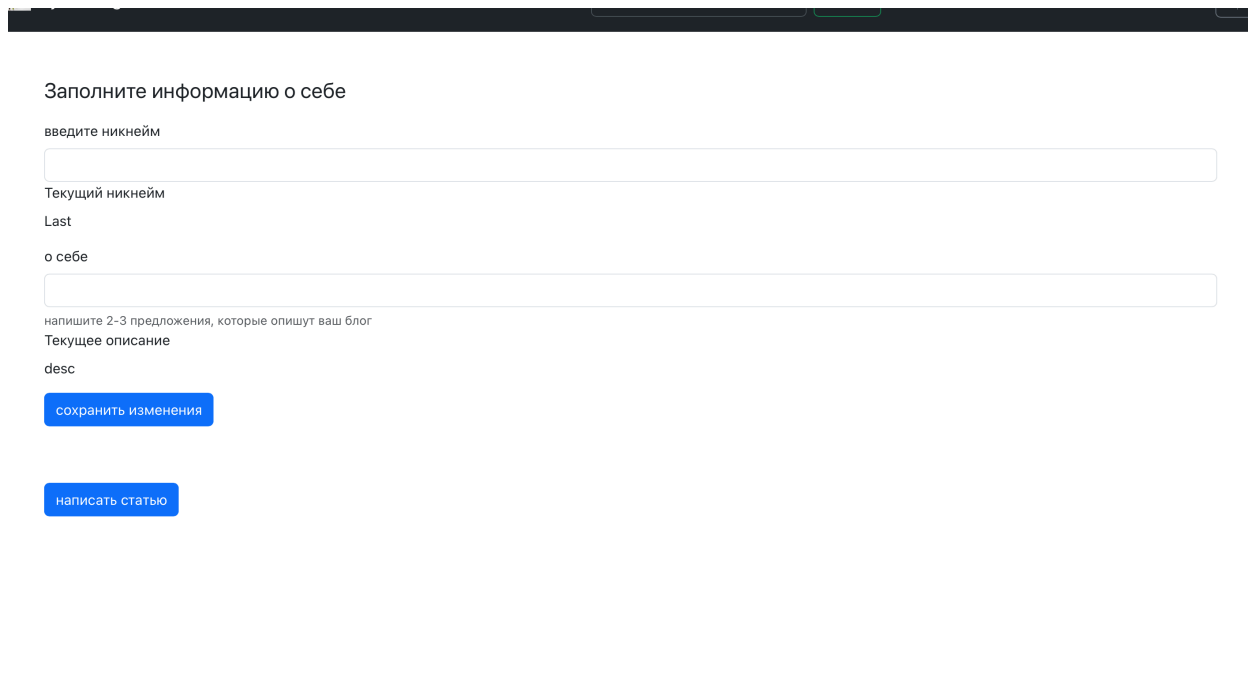


Рисунок 3 – страница личного кабинета в темной теме



Заполните информацию о себе

введите никнейм

Текущий никнейм

Last

о себе

напишите 2-3 предложения, которые опишут ваш блог

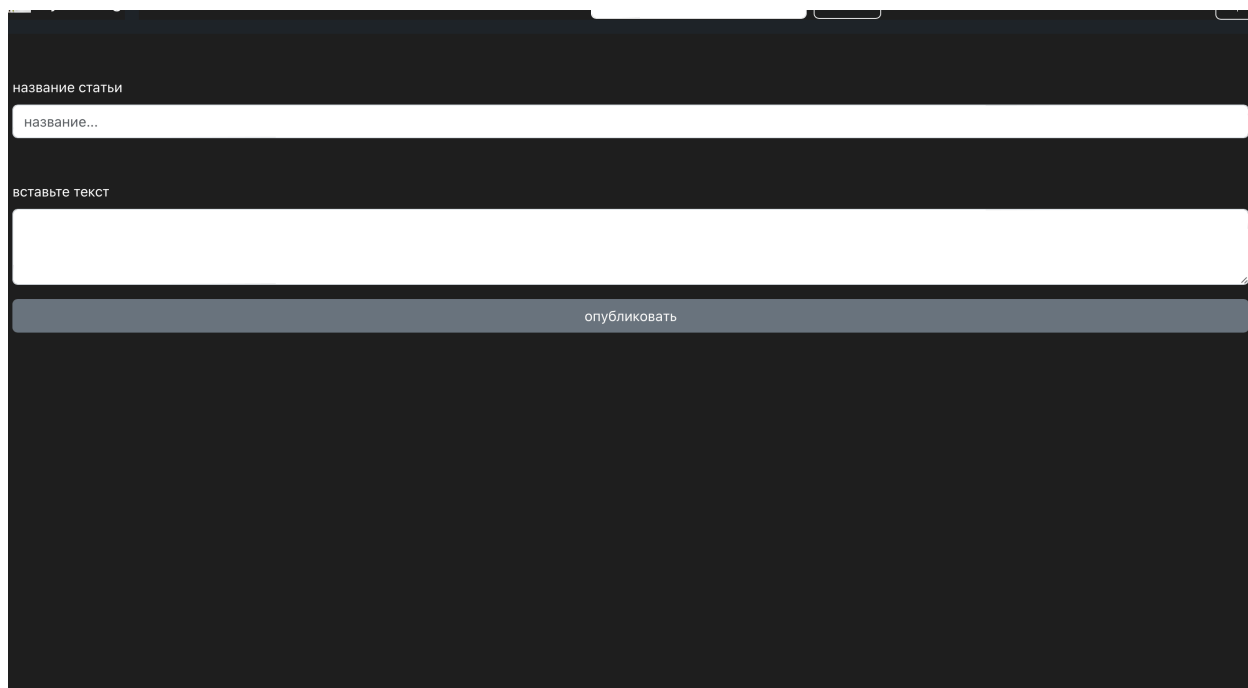
Текущее описание

desc

сохранить изменения

написать статью

*Рисунок 4 – страница личного кабинета в светлой теме*



название статьи

название...

вставьте текст

опубликовать

*Рисунок 5 - страница написания статьи в темной теме*



название статьи

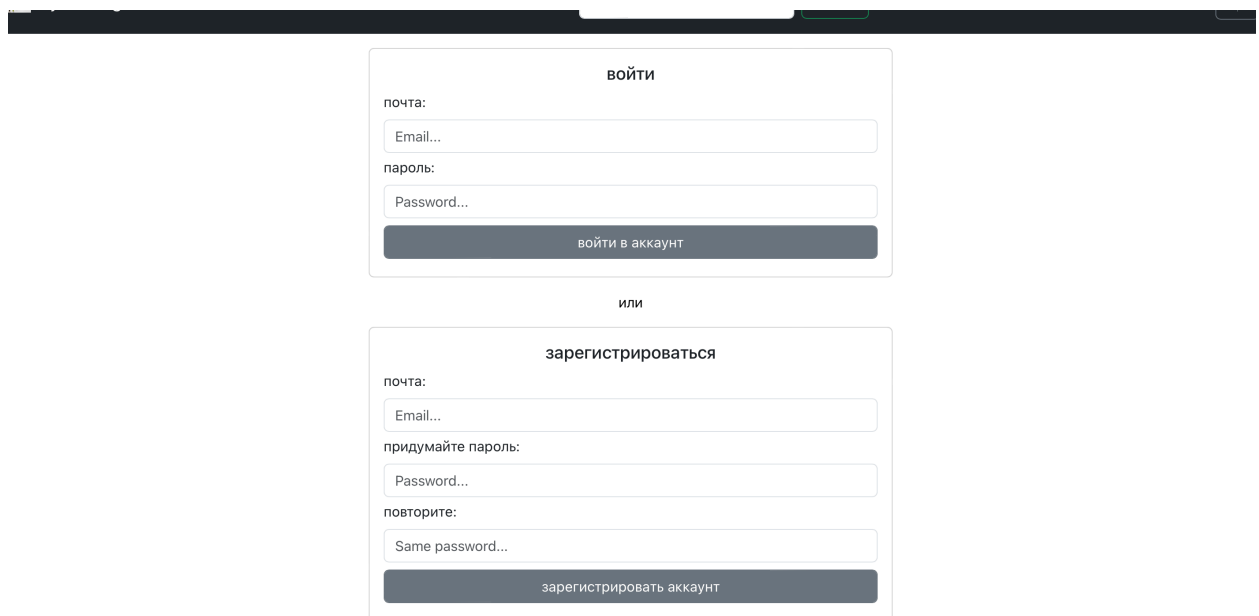
название...

вставьте текст

опубликовать

This screenshot shows the top part of a web interface for writing an article. It features a dark header bar at the top. Below it, there is a form with two input fields: one for the article title (labeled 'название статьи') and a larger one for the article content (labeled 'вставьте текст'). A dark button labeled 'опубликовать' (publish) is positioned at the bottom of the form.

*Рисунок 6 – страница написания статьи в светлой теме*



войти

почта:

Email...

пароль:

Password...

войти в аккаунт

или

зарегистрироваться

почта:

Email...

придумайте пароль:

Password...

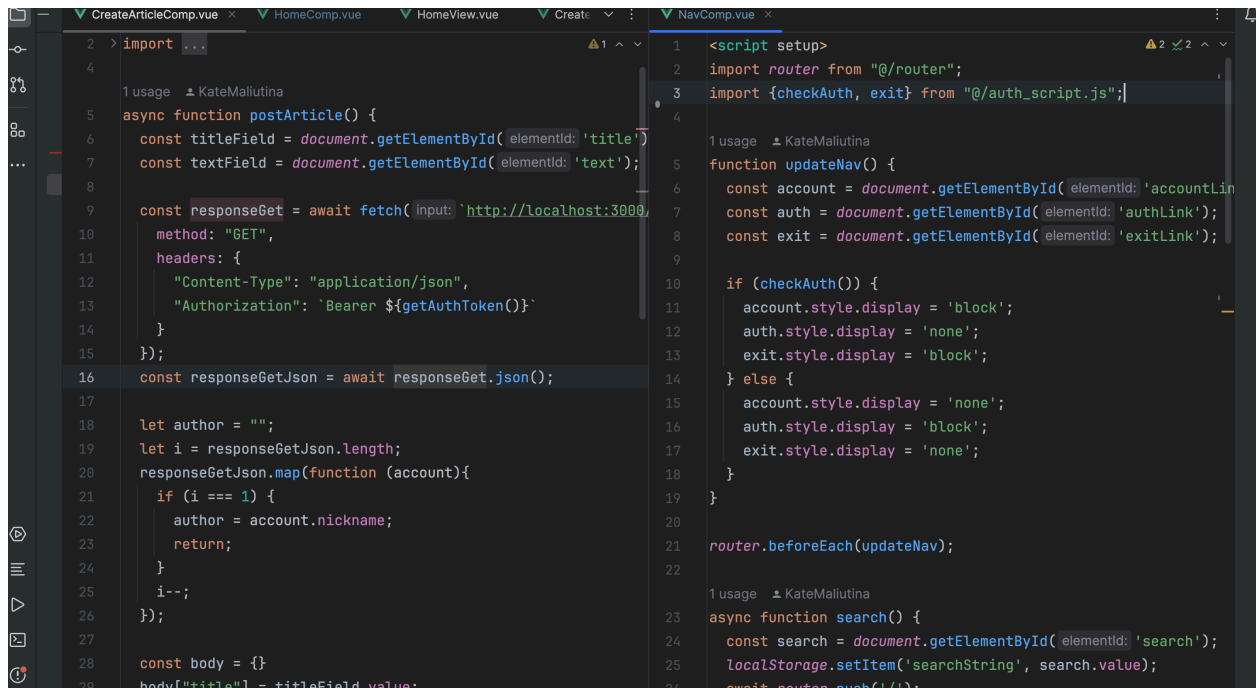
повторите:

Same password...

зарегистрировать аккаунт

This screenshot displays the login and registration section of the web interface. It is divided into two main parts. The top part is for login, titled 'войти', and includes fields for email ('почта: Email...') and password ('пароль: Password...'), followed by a 'войти в аккаунт' button. Below this is a separator 'или' (or). The bottom part is for registration, titled 'зарегистрироваться', and includes fields for email ('почта: Email...'), a new password ('придумайте пароль: Password...'), and a confirmation password ('повторите: Same password...'), followed by a 'зарегистрировать аккаунт' button.

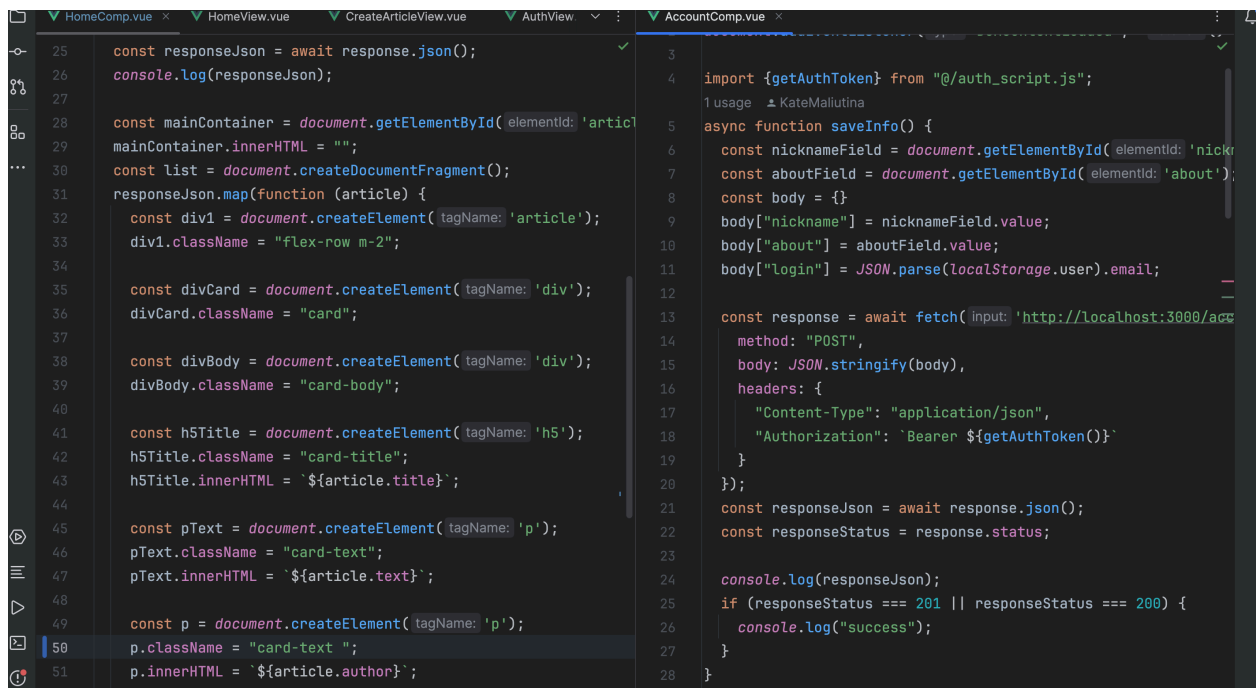
*Рисунок 7 – Страница авторизации в светлой теме*



```
2 > import ...
4
1 usage 1 KateMaliutina
5 async function postArticle() {
6   const titleField = document.getElementById('title');
7   const textField = document.getElementById('text');
8
9   const responseGet = await fetch(input: 'http://localhost:3000/
10     method: "GET",
11     headers: {
12       "Content-Type": "application/json",
13       "Authorization": 'Bearer ${getAuthToken()}'
14     }
15   });
16   const responseGetJson = await responseGet.json();
17
18   let author = "";
19   let i = responseGetJson.length;
20   responseGetJson.map(function (account){
21     if (i === 1) {
22       author = account.nickname;
23       return;
24     }
25     i--;
26   });
27
28   const body = {}
29   body["title"] = titleField.value;
```

```
1 <script setup>
2 import router from "@router";
3 import {checkAuth, exit} from "@auth_script.js";
4
1 usage 1 KateMaliutina
5 function updateNav() {
6   const account = document.getElementById('accountLink');
7   const auth = document.getElementById('authLink');
8   const exit = document.getElementById('exitLink');
9
10   if (checkAuth()) {
11     account.style.display = 'block';
12     auth.style.display = 'none';
13     exit.style.display = 'block';
14   } else {
15     account.style.display = 'none';
16     auth.style.display = 'block';
17     exit.style.display = 'none';
18   }
19 }
20
21 router.beforeEach(updateNav);
22
1 usage 1 KateMaliutina
23 async function search() {
24   const search = document.getElementById('search');
25   localStorage.setItem('searchString', search.value);
26   await router.push('/');
```

Рисунок 8 – создание статьи и навигация



```
25 const responseJson = await response.json();
26 console.log(responseJson);
27
28 const mainContainer = document.getElementById('article');
29 mainContainer.innerHTML = "";
30 const list = document.createDocumentFragment();
31 responseJson.map(function (article) {
32   const div1 = document.createElement('div');
33   div1.className = "flex-row m-2";
34
35   const divCard = document.createElement('div');
36   divCard.className = "card";
37
38   const divBody = document.createElement('div');
39   divBody.className = "card-body";
40
41   const h5Title = document.createElement('h5');
42   h5Title.className = "card-title";
43   h5Title.innerHTML = `${article.title}`;
44
45   const pText = document.createElement('p');
46   pText.className = "card-text";
47   pText.innerHTML = `${article.text}`;
48
49   const p = document.createElement('p');
50   p.className = "card-text";
51   p.innerHTML = `${article.author}`;
```

```
3
4 import {getAuthToken} from "@auth_script.js";
1 usage 1 KateMaliutina
5 async function saveInfo() {
6   const nicknameField = document.getElementById('nickname');
7   const aboutField = document.getElementById('about');
8   const body = {}
9   body["nickname"] = nicknameField.value;
10   body["about"] = aboutField.value;
11   body["login"] = JSON.parse(localStorage.user).email;
12
13   const response = await fetch(input: 'http://localhost:3000/ac
14     method: "POST",
15     body: JSON.stringify(body),
16     headers: {
17       "Content-Type": "application/json",
18       "Authorization": 'Bearer ${getAuthToken()}'
19     }
20   });
21   const responseJson = await response.json();
22   const responseStatus = response.status;
23
24   console.log(responseJson);
25   if (responseStatus === 201 || responseStatus === 200) {
26     console.log("success");
27   }
28 }
```

Рисунок 9 – отображение статей и аккаунта

## Выводы.

Сайт успешно мигрирован на Vue3, изучен router, внедрен json-server-auth, обновлены HTML-страницы и JavaScript-скрипты. Были получены навыки работы с упомянутыми инструментами и задачи на работу выполнены в полном объеме.