

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Лабораторная работа 3: разработка одностраничного веб-приложения (SPA) с использованием фреймворка Vue.js

Выполнил:

Мальшенко Александр

Группа: K33392

Проверил:

Добряков Д. И.

Санкт-Петербург
2023 г.

Оглавление

Задача (Вариант 10):.....	2
Ход выполнения работы.	2
Исправление недочетов.....	14
Вывод.....	18

Задача (Вариант 10):

Мигрировать ранее написанный сайт на фреймворк Vue.JS.

Ход выполнения работы.

Для начала создадим файлы для работы с api капсул ([src/api/capsules.js](#)) и пользователей ([src/api/users.js](#)). А также основной файл ([src/api/index.js](#)), в котором создаем экземпляры этих api.

Файл `api/capsules.js`:

```
class CapsulesApi {
  constructor(instance) {
    this.API = instance
  }

  getAllCapsules = async (searchString = '') => {
    if (!searchString) {
      return this.API({
        url: '/capsules'
      })
    } else {
      return this.API({
        url: `/capsules?title=${searchString}`
      })
    }
  }

  getOneCapsuleById = async (data) => {
    return this.API({
      url: `/capsules?id=${data}`
    })
  }

  getPublicCapsules = async (data) => {
    return this.API({
      url: '/capsules?capsuleAvailability=public'
    })
  }

  createCapsule = async (data) => {
    return this.API({
```

```

        method: 'POST',
        url: '/capsules',
        data,
        headers: {'Content-Type': 'application/json'}
      })
    }
  }
}

export default CapsulesApi

```

Файл api/users.js:

```

class UsersApi {
  constructor(instance) {
    this.API = instance
  }

  getAllUsers = async () => {
    return this.API({
      url: '/users'
    })
  }

  getUserById = async (data) => {
    return this.API({
      url: `/users?id=${data}`
    })
  }

  createUser = async (data) => {
    return this.API({
      method: 'POST',
      url: '/users',
      data,
      headers: {'Content-Type': 'application/json'}
    })
  }

  loginUser = async (data) => {
    return this.API({
      method: "POST",
      url: "/login",
      data,
      headers: {
        'Accept': 'application/json',
        'Content-Type': 'application/json'
      }
    })
  }
}

export default UsersApi

```

Далее в файле src/stores/index.js настраиваем и импортируем стейт-менеджер:

```

import { persist } from 'pinia-persists'
import { createPinia } from 'pinia'

```

```
const pinia = createPinia()

pinia.use(persist())

export default pinia
```

Создаем хранилища для работы с api капсул (src/stores/capsules.js) и пользователей (src/stores/users).

Файл stores/capsules.js:

```
import { defineStore } from "pinia"
import { capsulesApi } from "@/api"
import { usersApi } from "@/api";

const capsulesStore = defineStore('capsules', {
  state: () => ({
    capsules: [],
    capsule: {},
    sorting: "allCapsules"
  }),
  actions: {
    async loadCapsules(searchString = ''){
      if (!JSON.parse(localStorage.pinia_users)['user']){
        this.capsules
        const response = await
capsulesApi.getPublicCapsules(searchString)
        this.capsules = response.data
        return response

      } else {
        this.capsules
        const response = await
capsulesApi.getAllCapsules(searchString)
        this._loadLoginUserCapsules(response.data,
JSON.parse(JSON.parse(localStorage.pinia_users)['user']), (await
usersApi.getAllUsers())['data'])
        return response
      }
    },
    async createCapsule(data){
      const response = await capsulesApi.createCapsule(data)
      //this.capsules = response.data
      return response
    },
    async loadOneCapsuleById(data){
      const response = await capsulesApi.getOneCapsuleById(data)
      this.capsule = response.data[0]
      console.log(response)
      return response
    },
    _loadLoginUserCapsules(capsules, localUser, users){
      this.capsules = []
      for(const capsule of capsules){
```

```

        //don't show private not owner capsules
        if(capsule['capsuleAvailability'] === 'private' &&
capsule['userId'] !== localUser['id']){
            continue
        }

        if ((this.sorting === 'myCapsules' || this.sorting ===
'allCapsules') && capsule['userId'] === localUser['id']){
            this.capsules.push(capsule)
            continue
        }

        if (this.sorting === 'openedCapsules'){
            if(capsule['capsuleAvailability'] === 'public' &&
capsule['openingDate'] <= this._getCurrentDate()){
                this.capsules.push(capsule)
                continue
            }

            if(capsule['userId'] === localUser['id'] &&
capsule['openingDate'] <= this._getCurrentDate()){
                this.capsules.push(capsule)
                continue
            }

            if(capsule['capsuleAvailability'] === 'friendsOnly' &&
capsule['openingDate'] <= this._getCurrentDate()){
                for(const user of users){
                    if(capsule['userId'] === user.id &&
user.friendsList.includes(localUser['id'])){
                        this.capsules.push(capsule)
                        break
                    }
                }
            }
        }

        if(this.sorting === 'allCapsules') {
            if (capsule['capsuleAvailability'] === 'public') {
                this.capsules.push(capsule)
                continue
            }

            if (capsule['capsuleAvailability'] === 'friendsOnly') {
                for (const user of users) {
                    if (capsule['userId'] === user.id &&
user.friendsList.includes(localUser['id'])){
                        this.capsules.push(capsule)
                        break
                    }
                }
            }
        }
    },

    _getCurrentDate(){
        let today = new Date();
        const dd = String(today.getDate()).padStart(2, '0');
        const mm = String(today.getMonth() + 1).padStart(2, '0');
        //January is 0!
        const yyyy = today.getFullYear();

        today = yyyy + '-' + mm + '-' + dd;

```

```

        return today
    }
}
}))

export default capsulesStore

```

Файл stores/users.js:

```

import { defineStore } from "pinia"
import { usersApi } from "@api"
import stores from "@stores/index.js";
import users from "@api/users.js";

const usersStore = defineStore('users', {
  state: () => ({
    user: "", 3
    token: ""
  }),

  getters: {
    checkAuth: state => {
      if (localStorage.accessToken) {
        state.user = localStorage.user
        state.token = localStorage.accessToken
      }
      return (state.token !== "")
    }
  },

  actions: {
    async loadUsers() {
      const response = (await usersApi.getAllUsers())['data']
      const {accessToken, user} = response
      this.user = JSON.stringify(user)
      this.token = JSON.stringify(accessToken)
      return response
    },

    async createUser(data) {
      const response = (await usersApi.createUser(data))['data']
      const {accessToken, user} = response
      this.user = JSON.stringify(user)
      this.token = JSON.stringify(accessToken)
      return response
    },

    async loginUser(data) {
      const response = (await usersApi.loginUser(data))['data']
      const {accessToken, user} = response
      this.user = JSON.stringify(user)
      this.token = JSON.stringify(accessToken)
      return response
    }
  }
}))

export default usersStore

```

Далее создадим файл конфигурации роутера (src/router/index.js) и запишем в него массив со всеми путями, которые использует сайт:

```
import {createRouter, createWebHistory} from 'vue-router'

const router = createRouter({
  history: createWebHistory(import.meta.env.BASE_URL),

  //routers list
  routes: [
    { //home page
      path: '/timeCapsules',
      name: 'homePage',
      component: () => import('../views/homePage.vue')
    },
    { // profile page
      path: '/timeCapsules/profile',
      name: 'profilePage',
      component: () => import('../views/profilePage.vue')
    }
  ]
})

export default router
```

Теперь необходимо подключить все, что написали ранее в файле main.js:

```
import { createApp } from 'vue'

import App from '@App.vue'
import router from '@router'
import store from '@stores'
import 'bootstrap/dist/css/bootstrap.min.css'
import 'bootstrap'
import '@assets/main.css'

const app = createApp(App)

app.use(store)
app.use(router)
app.mount('#app')
```

Теперь можно приступить к написанию компонент для использования на сайте, так как при создании компонент использовался практически идентичный код с предыдущей лабораторной работы, покажу только несколько компонент.

Компонента основной открытой капсулы, отображаемой на странице (src/components/capsules/openedCapsuleCard.vue):

```
<script>
import modal from "@components/capsules/modalCapsule.vue";
import capsulesStore from "@stores/capsules.js";
import {Modal} from "bootstrap";

function getCurrentDate() {
  let today = new Date();
```

```

const dd = String(today.getDate()).padStart(2, '0');
const mm = String(today.getMonth() + 1).padStart(2, '0'); //January is 0!
const yyyy = today.getFullYear();

today = yyyy + '-' + mm + '-' + dd;
return today
}

export default {
  name: 'openedCapsuleCard',

  components: {modal},

  props: {
    id: {
      type: Number,
      required: true
    },
    title: {
      type: String,
      required: true
    },
    mainText: {
      type: String,
      required: false
    },
    creatorName: {
      type: String,
      required: true
    },
    creationDate: {
      type: String,
      required: true
    },
    openingDate: {
      type: String,
      required: true
    },
  },

  methods: {
    convertDate(date) {
      return date.substring(8) + '.' + date.substring(5, 7) + '.' +
date.substring(2, 4)
    },

    getCapsule() {
      this.capsulesState.loadOneCapsuleById(this.id);
      let capsuleModal = new Modal(document.getElementById('capsuleModal'));
      capsuleModal.show();
    },

    data() {
      return {
        capsulesState: capsulesStore(),
      }
    }
  }
}
</script>

<template>
  <div :data-capsule-id="id">
    <div class="card border-2 change-border-lightgreen">

```



```

<div class="card-header change-border-lightgreen change-bg-
verylightgreen"><h5 class="card-title">{{ title }}</h5>
</div>
<div class="card-body">
  <p class="card-text text-truncate">{{ mainText }}</p>
  <form>
    <div class="row justify-content-between gx-1">
      <button class="btn btn-my-lightgreen w-25" id="btnOpenCapsule"
type="button" @click="getCapsule">Open
      </button>
      <a href="#"
class="link-primary link-offset-2 link-underline-opacity-25
link-underline-opacity-100-hover text-truncate w-25 g-2">@{{
creatorName
}}</a>
    </div>
  </form>
</div>
<div class="card-footer">
  <div class="row justify-content-between gx-5">
    <p class="card-text w-50">Packing: <cite>{{
convertDate(creationDate) }}</cite></p>
    <p class="card-text w-50 text-success">Opening: <cite>{{
convertDate(openingDate) }}</cite></p>
  </div>
</div>
</div>
</div>
</template>

```

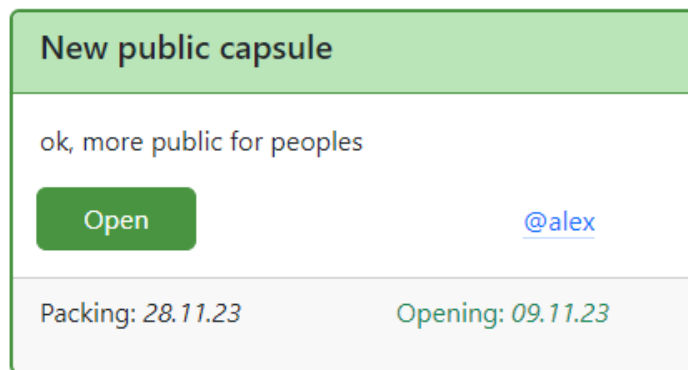


Рисунок 2.1. Отображение компоненты капсулы.

Компонента navbar`а зарегистрированного пользователя
(src/components/navbars/mainLoginUserNavbar.vue):

```

<script>
import ModalCreateCapsule from "@/components/modalCreateCapsule.vue";
import capsulesStore from "@/stores/capsules.js";

export default {
  name: 'mainLoginUserNavbar',
  components: {ModalCreateCapsule},
  data() {
    return {

```

```

        searchString: "",
        capsulesState: capsulesStore(),
    },
    computed: {
        updateCapsules() {
            this.capsulesState.loadCapsules(this.searchString)
        }
    },
}
</script>

<template>
    <nav class="navbar navbar-expand-lg bg-body-tertiary" id="navbar">
        <div class="container-fluid">
            <a class="navbar-brand" href="#">Time capsules</a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarSupportedContent"
                aria-controls="navbarSupportedContent" aria-expanded="false"
aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                    <li class="nav-item">
                        <a class="nav-link active" aria-current="page"
href="/timeCapsules">Home</a>
                    </li>
                    <li class="nav-item">
                        <button type="button" class="btn btn-my-lightgreen me-1 " data-
bs-toggle="modal"
                            data-bs-target="#createCapsuleModal">
                            Create capsule
                        </button>
                    </li>
                </ul>
                <form class="d-flex border-end border-2 me-2" role="search">
                    <input class="form-control me-1" type="search" placeholder="Search"
aria-label="Search" @input="updateCapsules"
id="searchString" v-model="searchString">
                    <button class="btn btn-my-lightgreen-outline me-2" type="button"
@click="updateCapsules">Search</button>
                </form>
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <button type="button" class="btn btn-my-lightgreen me-1"
onclick="location.href='/timeCapsules/profile';">
                            Account
                        </button>
                    </li>
                </ul>
            </div>
        </div>
    </nav>

    <modal-create-capsule/>
</template>

```

Рисунок 2.2. Отображение компоненты navbar`а

Теперь создадим представление главной страницы сайта
(src/views/homePage.vue):

```
<script>
import openedCapsuleCard from "@/components/capsules/openedCapsuleCard.vue"
import capsulesStore from "@/stores/capsules.js"
import MainUnregisterUserNavbar from
  "@/components/navbars/mainUnregisterUserNavbar.vue"
import MainRadioButtons from "@/components/auth/mainRadioButtons.vue"
import usersStore from "@/stores/users.js";
import MainLoginUserNavbar from
  "@/components/navbars/mainLoginUserNavbar.vue";
import ClosedCapsuleCard from "@/components/capsules/closedCapsuleCard.vue";
import Modal from "@/components/capsules/modalCapsule.vue";
import CapsuleModal from "@/components/capsules/modalCapsule.vue";

export default {
  name: 'homePage',
  components: {
    CapsuleModal,
    Modal,
    ClosedCapsuleCard, MainLoginUserNavbar, MainRadioButtons,
    MainUnregisterUserNavbar, openedCapsuleCard
  },
  computed: {
    checkLogin() {
      return this.userState.checkAuth
    }
  },
  methods: {
    async createCard() {
      await this.capsulesState.createCapsule(this.form)
      await this.capsulesState.loadCapsules()
      this.$refs.capsuleForm.reset()
    },
    getCurrentDate() {
      let today = new Date();
      const dd = String(today.getDate()).padStart(2, '0');
      const mm = String(today.getMonth() + 1).padStart(2, '0'); //January is
0!
      const yyyy = today.getFullYear();

      today = yyyy + '-' + mm + '-' + dd;
      return today
    },
  },
  mounted() {
    this.capsulesState.loadCapsules()
  },
  data() {
    return {
      userState: usersStore(),
      capsulesState: capsulesStore(),
    }
  }
}
</script>

<template>
  <header>
```

```

<div v-if="checkLogin">
  <main-login-user-navbar/>
</div>
<div v-else>
  <main-unregister-user-navbar/>
</div>
</header>

<main>
  <main-radio-buttons/>

  <div class="container">
    <div class="row" id="capsules">
      <div class="col-12 col-xl-4 col-md-6 col-sm-6 mb-3 mb-sm-0 mt-3" v-
for="capsule in this.capsulesState.capsules"
      :key="capsule.id">
        <div v-if="getCurrentDate() >= capsule.openingDate">
          <opened-capsule-card :id="capsule.id" :title="capsule.title"
:main-text="capsule.text"
          :creator-name="capsule.creatorName"
:creation-date="capsule.creationDate"
          :opening-date="capsule.openingDate"/>
        </div>
        <div v-else>
          <closed-capsule-card :id="capsule.id" :title="capsule.title"
:main-text="capsule.text"
          :creator-name="capsule.creatorName"
:creation-date="capsule.creationDate"
          :opening-date="capsule.openingDate"/>
        </div>
      </div>
    </div>
  </div>
</main>

<capsule-modal/>
</template>

```

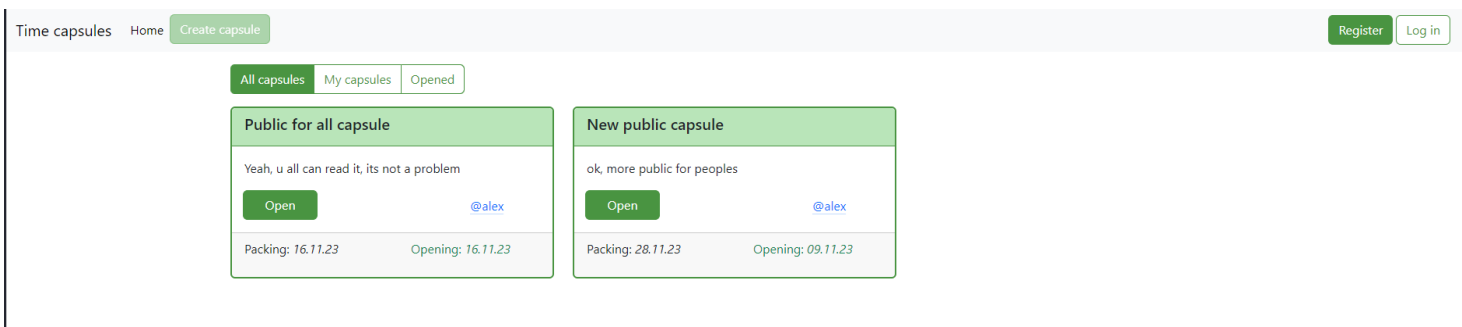


Рисунок 2.3. Главная страница сайта

Так же представление страницы с профилем пользователя
([src/views/profilePage.vue](#)):

```

<script>
import {defineComponent} from "vue";
import usersStore from "@stores/users.js";
import ProfileNavbar from "@components/navbars/profileNavbar.vue";

```

```

import ProfileInfo from "@/components/profileInfo.vue";

export default defineComponent({
  name: "profilePage",
  components: {ProfileInfo, ProfileNavbar},

  data() {
    return {
      userState: usersStore()
    }
  },


  methods: {
    getUser() {
      return JSON.parse(localStorage.user)
    }
  },

  mounted() {
    this.userState.loadUsers()
  },
})
</script>

<template>
  <header>
    <nav class="navbar bg-body-tertiary fixed-top">
      <div class="container-fluid" id="profileNavbar">
        <profile-navbar :username="getUser().username"/>
      </div>
    </nav>
  </header>

  <main>
    <div class="container py-5 mt-5">
      <div class="row">
        <profile-info :address="getUser().address" :phone-
number="getUser().phoneNumber" :email="getUser().email"
:full-name="getUser().fullUsername" :extra-
info2="getUser().extraProfileInfo1"
:extra-info1="getUser().extraProfileInfo2"
:username="getUser().username"/>
      </div>
    </div>
  </main>
</template>

```



alex

Full name

Email cazas123toy@gmail.com

Phone

Address

Number of capsules

5

Рисунок 2.4. Страница с профилем пользователя.

Исправление недочетов.

1) Поиск:

Добавил строку с кнопкой для поиска по капсулам в навбар. Поиск работает по **не полному** соответствию, так же ищет не только по заголовку карточек, но и по тексту в них, автору или же дате.

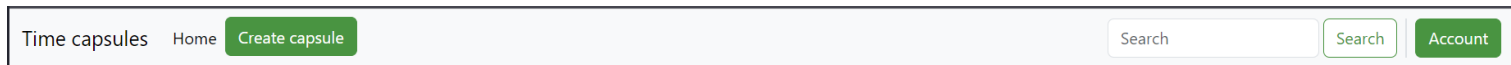


Рисунок 3.1. Строка поиска в навбаре

Примеры:

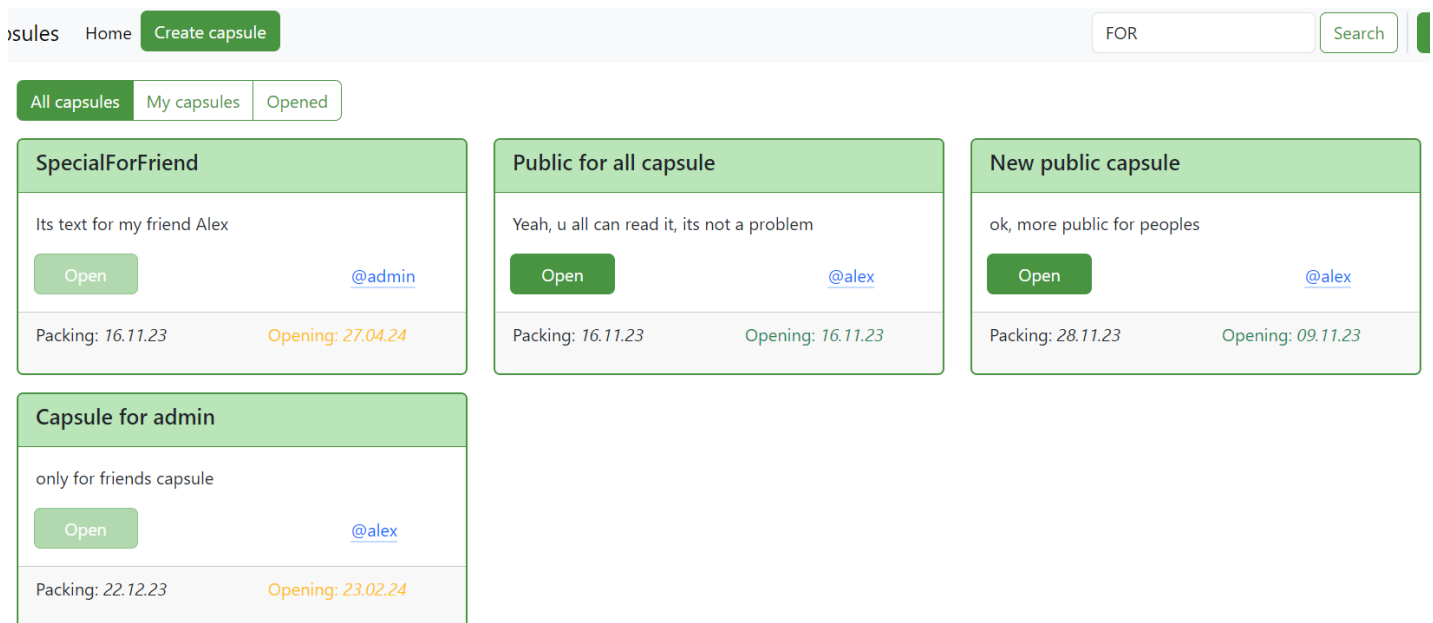


Рисунок 3.2. Совпадения в заголовке и тексте капсул (регистр не важен).

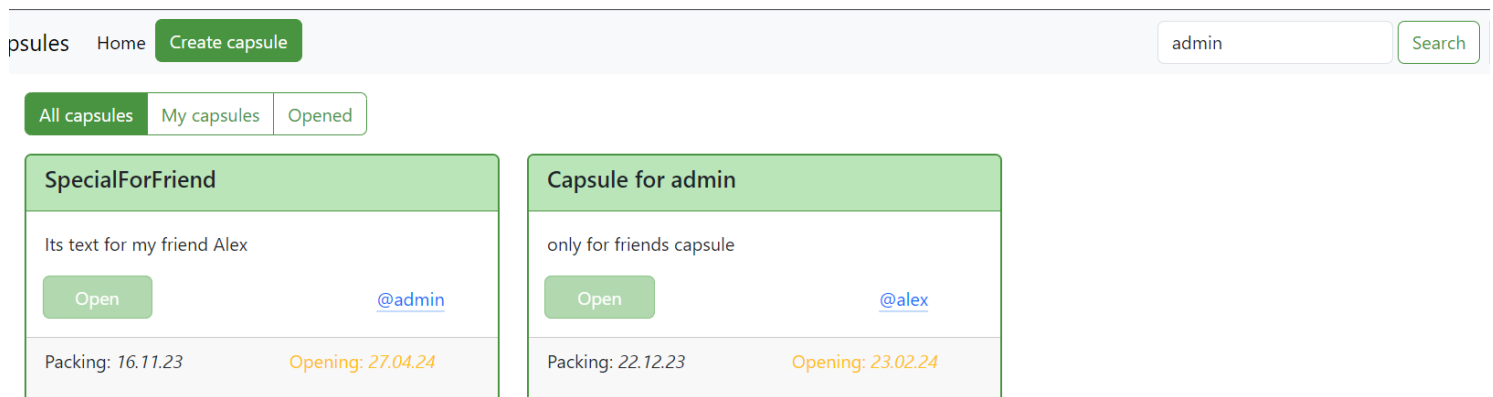


Рисунок 3.3. Совпадения по автору и заголовку капсулы.

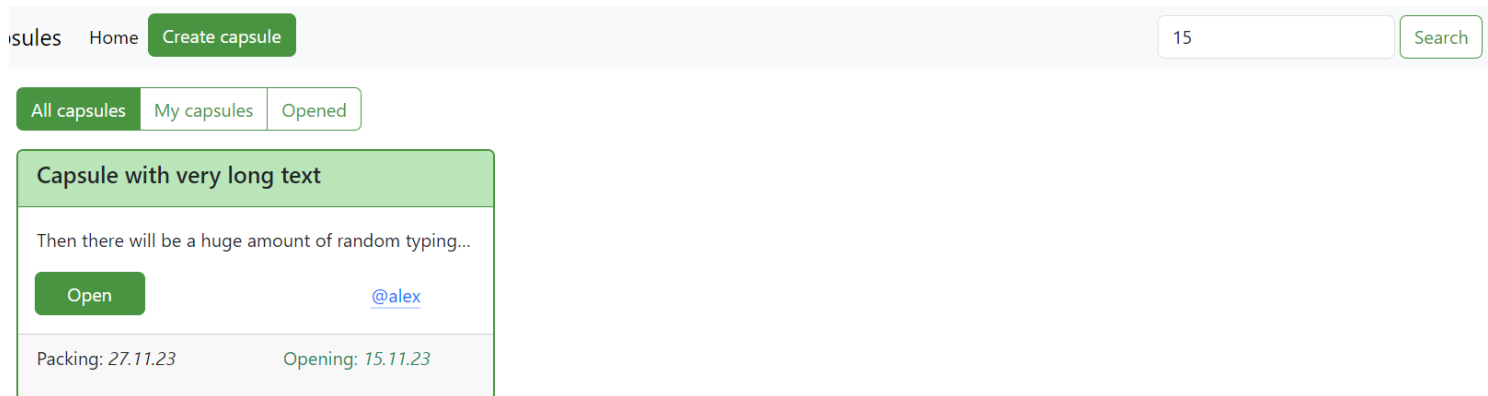


Рисунок 3.4. Совпадения по дате.

2) Composable:

Не знал, где использовать composable в моем варианте сайта, поэтому написал composable функцию для отслеживания передвижения мышки, и, если пользователь в профиле наводится на свою аватарку, появляется кнопка для редактирования.

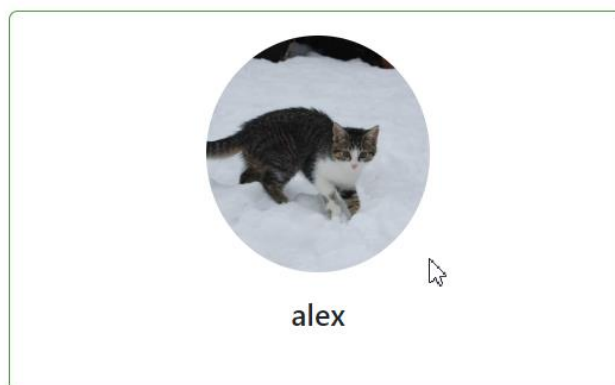


Рисунок 3.5. Курсор не на аватарке.

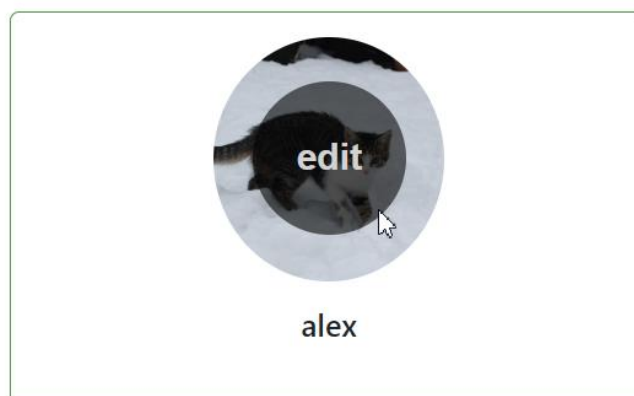


Рисунок 3.6. Курсор на аватарке.

Composable функция в файле «src/profileAvatarMouseCheck.js»:

```
import { ref, onMounted, onUnmounted } from 'vue'

export function mouseOnAvatar() {
  const onAvatar = ref(false)

  function update(event) {
    const xPos = event.pageX
    const yPos = event.pageY

    if (mouseInCircle(xPos, yPos)) {
      onAvatar.value = true
      return 0
    }

    onAvatar.value = false
  }

  function mouseInCircle(x, y) {
    const xMidAv = 295
    const yMidAv = 188
    const maxDist = 70
    return (Math.sqrt(Math.pow((xMidAv - x), 2) + Math.pow((yMidAv - y), 2))) < maxDist
  }

  onMounted(() => window.addEventListener('mousemove', update))
  onUnmounted(() => window.removeEventListener('mousemove', update))

  return { onAvatar }
}
```

Использование в компоненте «src/components/profileInfo.vue»:

```
<script setup>
  import {mouseOnAvatar} from "@/profileAvatarMouseCheck.js";

  const {onAvatar} = mouseOnAvatar()
</script>
```

```

<div class="text-on-img" v-if="onAvatar">edit</div>
```

```
<style>
div#text-on-img {
  position: relative;
}.text-on-img {
  position: absolute;
  top: 17.5%;
  left: 38%;
  color: lightgray;
  width: 100px;
  height: 100px;
  background: #00000094;
  padding: 10px;
  display: flex;
  align-items: center;
  text-align: center;
```



```
justify-content: center;
font: bold 24px/34px Helvetica, Sans-Serif;
border-radius: 50%;
user-select: none;
}
</style>
```

Вывод.

В ходе выполнения данной лабораторной работы я успешно мигрировал свой сайт на Vue.JS, реализовав работу с `api`, создав роутеры и разделив прошлые элементы сайта на компоненты.