

Portfolio Vue

Задание

Разработка одностраничного веб-приложения (SPA) с использованием фреймворка Vue.JS

Ход работы

Страницы входа и регистрации:

Для управления доступом к личному кабинету пользователя были созданы страницы входа и регистрации.

Вход

Имя пользователя или Email:

Пароль:

Войти

Форма для входа

```
<template>
  <div class="container mt-5">
    <header>
      <h1 class="text-center">Вход</h1>
    </header>
    <form @submit.prevent="handleLoginFormSubmit" class="col-md-4 mx
      <div class="mb-3">
        <label for="username" class="form-label">Email:</label>
        <input v-model="email" type="text" id="username" name="email
      </div>
      <div class="mb-3">
```

```

    <label for="password" class="form-label">Пароль:</label>
    <input v-model="password" type="password" id="password" name
  </div>
  <button type="submit" class="btn btn-primary mt-3">Войти</butt

  <div v-if="error" class="alert alert-danger mt-3" role="alert"
    {{ error }}
  </div>
</form>
<div class="text-center">
  <p class="mt-3">
    <router-link to="/register">Создать аккаунт</router-link>
  </p>
</div>
</div>
<theme-switcher></theme-switcher>
</template>

<script>
import ThemeSwitcher from "@components/ThemeSwitcher.vue";
import 'bootstrap/dist/css/bootstrap.min.css';

export default {
  components: {
    ThemeSwitcher
  },
  data() {
    return {
      email: '',
      password: '',
      error: null,
      themes: ['light', 'dark', 'auto'],
      currentThemeIndex: 0,
    };
  },
  methods: {
    async sendLoginData(loginData) {
      try {
        const response = await fetch('http://localhost:3000/login',
          method: 'POST',
          body: JSON.stringify(loginData),
          headers: {
            'Content-Type': 'application/json',

```

```

    },
  });

  const responseJson = await response.json();
  this.handleLoginResponse(responseJson);
} catch (error) {
  console.error('Error:', error);
  this.displayErrorAlert();
}
},
handleLoginResponse(responseJson) {
  if (responseJson.accessToken) {
    localStorage.accessToken = responseJson.accessToken;
    localStorage.user = JSON.stringify(responseJson.user);
    this.$router.push( '/profile' )
  } else {
    this.displayErrorAlert();
  }
},
displayErrorAlert() {
  this.error = 'Неправильный email или пароль. Пожалуйста, попробуйте еще раз.';
  setTimeout(() => {
    this.error = null;
  }, 3000);
},
handleLoginFormSubmit() {
  const loginData = {
    email: this.email,
    password: this.password,
  };
  this.sendLoginData(loginData);
},
},
};
</script>

<style scoped>
@keyframes fadeIn {
  from { opacity: 0; }
  to { opacity: 1; }
}

@keyframes fadeOut {

```

```
from { opacity: 1; }  
to { opacity: 0; }  
}  
</style>
```

Регистрация

Имя:

Фамилия:

Email:

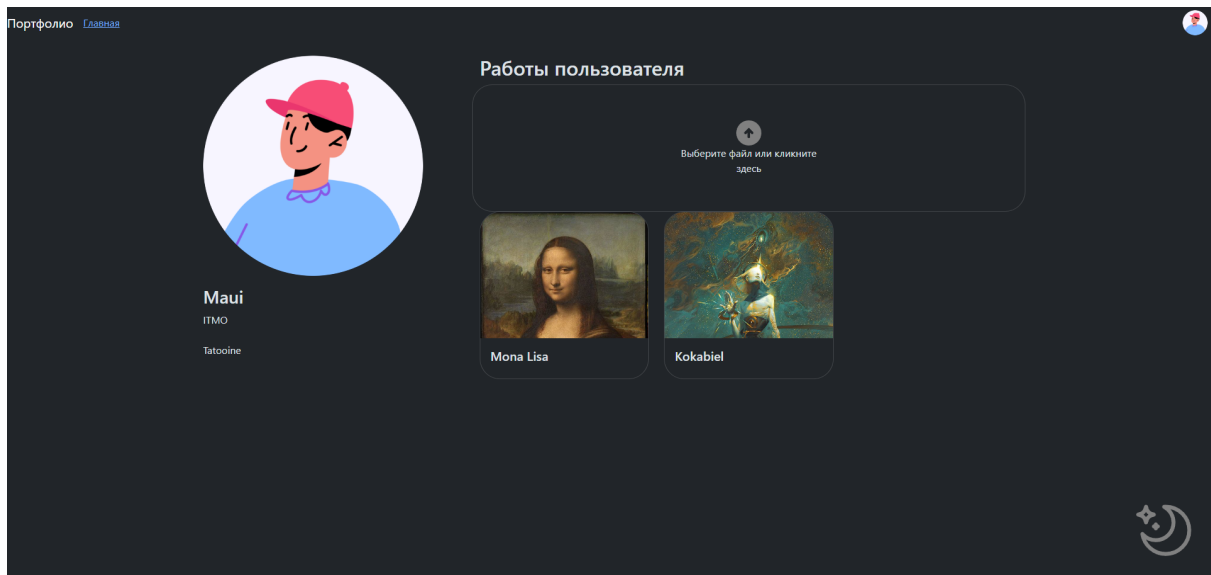
Пароль:

Подтвердите пароль:

Зарегистрироваться

Форма для регистрации

Экран профиля пользователя



```
<template>
  <nav class="navbar navbar-expand-lg navbar-light">
    <a class="navbar-brand" href="#">Портфолио</a>

    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav ml-auto">
        <li class="nav-item">
          <router-link to="/gallery">Главная</router-link>
        </li>
      </ul>
    </div>
    <div class="dropdown">
      <img src="" alt="Фото профиля" class="img-fluid rounded-circle"
        style="width: 40px; height: 40px; margin-right: 10px;"
        id="avatar-small" @click="toggleDropdown" width="40"
        height="40">

      <ul class="dropdown-menu" v-show="isDropdownOpen">
        <li><a class="dropdown-item" @click="logout">Выйти</a></li>
      </ul>
    </div>
  </nav>

  <div class="container mt-4">
    <div class="row">
      <div class="col-md-4">
```

```

<div class="row" id="full_avatar"></div>

<h3 id="user-name">{{ userData.firstname }}</h3>
<div class="color-fg-muted mt-2 d-flex flex-items-center"
id="company-name-container"
v-show="userData.company">

    <p id="company-name">{{ userData.company }}</p>
</div>
<div class="color-fg-muted mt-2 d-flex flex-items-center"
id="location-container" v-show="userData.location">

    <p id="location">{{ userData.location }}</p>
</div>
<div id="edit-profile-form" style="display: none;">
    <form @submit.prevent="submitProfileForm">
        <div class="form-group">
            <label for="new-username">Новое имя пользователя:</label>
            <input v-model="newUsername" type="text" class="form-control" id="new-username" name="new-username">
        </div>
        <div class="color-fg-muted mt-2 d-flex flex-items-center" v-show="userData.company">
            <!-- 
            <input class="ml-2 form-control flex-auto input-sm" placeholder="Company" aria-label="Company"
name="company" v-model="newCompany">
        </div>
        <div class="color-fg-muted mt-2 d-flex flex-items-center" v-show="userData.location">
            <!-- 
            <input class="ml-2 form-control flex-auto input-sm" placeholder="Location" aria-label="Location"
name="location" v-model="newLocation">
        </div>
        <button type="submit" class="btn btn-success mt-3">Сохранить</button>
        <button type="button" class="btn btn-outline-secondary

```

```

mt-3" @click="cancelEdit">Отменить</button>
    </form>
  </div>

</div>

<div class="col-md-8">
  <h2>Работы пользователя</h2>
  <div class="row">
    <div class="card" style="cursor: pointer;" data-bs-toggl
e="modal" data-bs-target="#addImageModal">
      <div
        style="border-radius: 32px; height: 200px; displa
y: flex; flex-direction: column; align-items: center; justify-cont
ent: center;">
        <div style="border-radius: 32px; max-width: 220px; t
ext-align: center;">
          
          <div>Выберите файл или кликните здесь</div>
        </div>
      </div>
    <div class="col-md-4 mb-3" v-for="(image, index) in imag
es" :key="index">
      <div class="card" style="cursor: pointer;" @click="ope
nModal(image.imageFile)">
        
        <div class="card-body">
          <h5 class="card-title">{{ image.imageTitle }}</h5>
        </div>
      </div>
    </div>
  </div>
</div>
</div>

```

```

<!-- Модальное окно для увеличения изображения -->
<div class="modal fade" id="imageModal" tabindex="-1" aria-labelledby="imageModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="imageModalLabel">Увеличенное изображение</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        
      </div>
    </div>
  </div>
</div>

<div class="modal fade" id="addImageModal" tabindex="-1" aria-labelledby="addImageModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="addImageModalLabel">Добавить изображение</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        <form enctype="multipart/form-data" @submit.prevent ref="imageForm">
          <div class="mb-3">
            <label for="imageFile" class="form-label">Выберите файл изображения (jpg/png):</label>
            <input type="file" class="form-control" id="imageFile" name="imageFile" accept=".jpg, .jpeg, .png">
          </div>
          <div class="mb-3">
            <label for="imageTitle" class="form-label">Название изображения:</label>
            <input type="text" class="form-control" id="imageTit

```



```

le" name="imageTitle">
    </div>
    <button @click="uploadImage" type="submit" class="btn
btn-primary" data-bs-dismiss="modal">Добавить</button>
    </form>
  </div>
</div>
</div>
</div>

<theme-switcher></theme-switcher>
</template>

<script>
import ic_add from "@/images/ic_add.svg"
import ThemeSwitcher from "@/components/ThemeSwitcher.vue";
import useUserData from "@/composables/useUserData.js";
import useModal from "@/composables/useModal.js";
import {onMounted} from "vue";

export default {
  components: {ThemeSwitcher},
  setup() {
    const {
      userData,
      images,
      selectedImage,
      fillUserInfo,
      loadImages,
      updateProfile,
    } = useUserData();

    const {
      isImageModalOpen,
      isAddImageModalOpen,
      openImageModal,
      closeImageModal,
      openAddImageModal,
      closeAddImageModal,
    } = useModal();

```

```

    onMounted(() => {
      fillUserInfo();
      loadImages();
    });
    const getUserId = () => {
      const userData = localStorage.user ? JSON.parse(localStorage.user) : null;
      return userData ? userData.id : null;
    };

    return {
      ic_add,
      getUserId,
      userData,
      images,
      selectedImage,
      fillUserInfo,
      loadImages,
      updateProfile,
      isImageModalOpen,
      isAddImageModalOpen,
      openImageModal,
      closeImageModal,
      openAddImageModal,
      closeAddImageModal,
    };
  },
  data() {
    return {
      userData: {},
      images: [],
      selectedImage: "",
    };
  },
  getUserId() {
    const userData = localStorage.user ? JSON.parse(localStorage.user) : null;
    return userData ? userData.id : null;
  },
  methods: {
    openModal(imageFile) {
      this.selectedImage = imageFile;
    }
  }
}

```

```

    },

    async uploadImage(event) {
      const form = this.$refs.imageForm;
      const formData = new FormData(form);
      formData.append('userId', this.getUserId);
      try {
        const response = await fetch('http://localhost:3000/images', {
          method: 'POST',
          body: formData,
        });

        if (response.ok) {
          console.log('Файл успешно загружен!');
        } else {
          console.error('Ошибка загрузки файла:', response.status);
        }
      } catch (error) {
        console.error('Произошла ошибка:', error);
      }
    },

    toggleDropdown() {
      const dropdown = document.getElementById('avatar-dropdown');
      const avatar = document.getElementById('avatar-small');
      if (dropdown.style.display === 'block') {
        dropdown.style.display = 'none';
        document.querySelector('.dropdown').classList.remove('active');
      } else {
        dropdown.style.display = 'block';
        document.querySelector('.dropdown').classList.add('active');
      }
    },

    handleLogoutClick() {
      localStorage.clear();
      window.location.href = 'login.html';
    },
  },

```

```

    },
  },
};
</script>
<style scoped>
.card {
  border-radius: 32px;
  overflow: hidden;
}

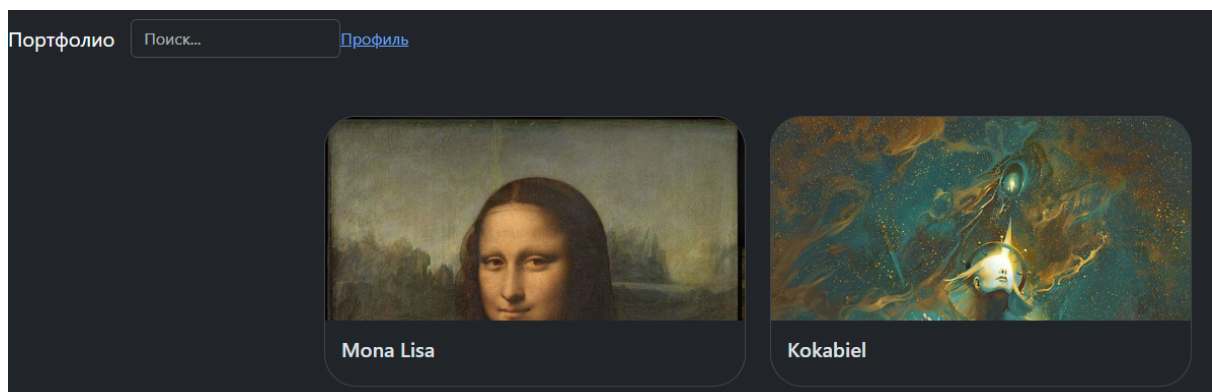
.card-img-top {
  object-fit: cover;
  object-position: center top;
  height: 200px;
  transition: transform 0.3s;
}

.card:hover .card-img-top {
  transform: scale(1.1);
}

.modal-image {
  max-width: 100%;
  max-height: 80vh;
  width: auto;
  height: auto;
  object-fit: contain;
}
</style>

```

Экран с работами всех пользователей



```

<!-- Gallery.vue -->

<template>
  <div>
    <nav class="navbar navbar-expand-lg navbar-light">
      <a class="navbar-brand" href="#">Портфолио</a>
      <form class="form-inline my-2 my-lg-0">
        <input id="searchInput" class="form-control mr-2 mr-sm-2"
type="search" placeholder="Поиск..." v-model="searchQuery"
        @input="searchGallery">
      </form>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav ml-auto">
          <li class="nav-item">
            <router-link to="/profile">Профиль</router-link>
          </li>
        </ul>
      </div>

      <div class="dropdown">
        <img src="" alt="Фото профиля" class="img-fluid rounded-ci
rcle"
          style="width: 40px; height: 40px; margin-right: 10p
x;" id="avatar-small" @click="toggleDropdown" width="40"
          height="40">

        <ul class="dropdown-menu" v-show="isDropdownOpen">
          <li><a class="dropdown-item" @click="logout">Выйти</a></
li>
        </ul>
      </div>
    </nav>

    <div class="container mt-5">
      <div class="row" id="gallery">
        <div class="col-md-4 mb-3" v-for="(image, index) in filter
edGallery" :key="index">
          <div class="card" style="cursor: pointer;" @click="openM
odal(image.imageFile)">
            
    l">
        <div class="card-body">
            <h5 class="card-title">{{ image.imageTitle }}</h5>
        </div>
    </div>
</div>
</div>
</div>

<!-- Модальное окно для увеличения изображения -->
<div class="modal fade" id="imageModal" tabindex="-1" aria-labelledby="imageModalLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="imageModalLabel">Увеличенн
ое изображение</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body">
                
            </div>
        </div>
    </div>
</div>
</div>
</div>

<theme-switcher></theme-switcher>

</template>

<script>
import ThemeSwitcher from "@components/ThemeSwitcher.vue";

export default {
    components: {ThemeSwitcher},
    data() {
        return {
            galleryData: [],
            selectedImage: "",

```

```

        searchQuery: "",
    };
},
mounted() {
    this.initializeUserStatus();
    this.loadGallery();
    this.fillUserInfo();
},
computed: {
    filteredGallery() {
        return this.galleryData.filter(image => image.imageTitle.toLowerCase().includes(this.searchQuery.toLowerCase()));
    },
},
methods: {
    getUserId() {
        const userData = localStorage.user ? JSON.parse(localStorage.user) : null;
        return userData ? userData.id : null;
    },
    openModal(imageFile) {
        this.selectedImage = imageFile;
    },
    initializeUserStatus() {
        const userData = localStorage.user ? JSON.parse(localStorage.user) : null;
        if (userData) {
            const avatarSmall = document.getElementById('avatar-small');
            avatarSmall.src = `https://avatar.iran.liara.run/public/${userData.avatar}`;
        }
    },
    async loadGallery() {
        const url = `http://localhost:3000/664/images`;

        const response = await fetch(url, {
            headers: {
                'Authorization': `Bearer ${localStorage.accessToken}`
            }
        });
        const responseJson = await response.json();
    }
}

```

```

    this.galleryData = responseJson.map(image => ({
      id: image.id,
      imageFile: image.imageFile,
      imageTitle: image.imageTitle,
    }));
  },
  fillUserInfo() {
  },
  searchGallery() {
  },
},
};
</script>

<style scoped>
.card {
  border-radius: 32px;
  overflow: hidden;
}

.card-img-top {
  object-fit: cover;
  object-position: center top;
  height: 200px;
  transition: transform 0.3s;
}

.card:hover .card-img-top {
  transform: scale(1.1);
}

.modal-image {
  max-width: 100%;
  max-height: 80vh;
  width: auto;
  height: auto;
  object-fit: contain;
}
</style>

```

Так же были использованы composables

useUserData.js

```
// useUserData.js
import { ref } from 'vue';

export default function useUserData() {
  const userData = ref({});
  const images = ref([]);
  const selectedImage = ref('');

  const getUserId = () => {
    const userDataFromLocalStorage = localStorage.user ? JSON.
parse(localStorage.user) : null;
    return userDataFromLocalStorage ? userDataFromLocalStorag
e.id : null;
  };

  const fillUserInfo = () => {
    const userDataFromLocalStorage = localStorage.user ? JSON.
parse(localStorage.user) : null;

    if (userDataFromLocalStorage) {
      userData.value = userDataFromLocalStorage;
    }
  };

  const loadImages = async () => {
    const userId = this.getUserId();
    const url = `http://localhost:3000/664/images`;

    const response = await fetch(url, {
      headers: {
        'Authorization': `Bearer ${localStorage.accessToken}`
      }
    });
    const responseJson = await response.json();

    console.log('response', responseJson);

    this.images = [];

    const filteredImages = responseJson.filter(image => image.
```

```

userId.toString() === userId.toString());
    console.log('response filteredImages', filteredImages);

    for (const image of filteredImages) {
        this.images.push({
            id: image.id,
            imageFile: image.imageFile,
            imageTitle: image.imageTitle,
        });
    }
};

const updateProfile = async (data) => {
    const token = localStorage.accessToken;
    const userId = getUserId();

    const requestOptions = {
        method: 'PATCH',
        headers: {
            'Content-Type': 'application/json',
            'Authorization': `Bearer ${token}`
        },
        body: JSON.stringify(data)
    };

    try {
        const response = await fetch(`http://localhost:3000/users/${userId}`, requestOptions);

        if (response.ok) {
            const updatedData = await response.json();

            const userDataFromLocalStorage = localStorage.user
            ? JSON.parse(localStorage.user) : null;
            if (userDataFromLocalStorage) {
                userDataFromLocalStorage.firstname = updatedData.firstname;
                userDataFromLocalStorage.company = updatedData.company;
                userDataFromLocalStorage.location = updatedData.location;
                localStorage.user = JSON.stringify(userDataFromLocalStorage);
            }
        }
    }
};

```

```

        fillUserInfo();
    }
    } else {
        throw new Error('Ошибка при обновлении профиля');
    }
    } catch (error) {
        console.error('Произошла ошибка при обновлении профиля:', error);
    }
    };

    return {
        userData,
        images,
        selectedImage,
        fillUserInfo,
        loadImages,
        updateProfile,
    };
}

```

Компонент смены темы

Был добавлен компонент, который позволяет менять тему сайта. Есть три режима темная, светлая и системная

ThemeSwitcher.vue

```

<template>
  <div class="bottom-right-container">
    
  </div>
</template>

<script>
import ic_sun from "@images/ic_sun.svg"
import ic_moon_sun from "@images/ic_moon_sun.svg"
import ic_moon from "@images/ic_moon.svg"

export default {

```

```

setup() {
  return {
    ic_sun,
    ic_moon_sun,
    ic_moon
  };
},
data() {
  return {
    themes: ['light', 'dark', 'auto'],
    currentThemeIndex: 0,
    themeImages: [ic_sun, ic_moon, ic_moon_sun],
    currentTheme: 'light',
  };
},
computed: {
  themeIcon() {
    return this.themeImages[this.currentThemeIndex];
  },
},
methods: {
  toggleTheme() {
    this.currentThemeIndex = (this.currentThemeIndex + 1) % this.themes.length;
    this.currentTheme = this.themes[this.currentThemeIndex];
    this.setStoredTheme(this.currentTheme);
    this.setTheme(this.currentTheme);
  },
  setStoredTheme(theme) {
    localStorage.theme = theme;
  },
  setTheme(theme) {
    if (theme === 'auto' && window.matchMedia('(prefers-color-scheme: dark)').matches) {
      document.documentElement.setAttribute('data-bs-theme', 'dark');
    } else {
      document.documentElement.setAttribute('data-bs-theme', theme);
    }
  },
},
created() {

```

```

    const storedTheme = localStorage.theme;
    this.currentTheme = storedTheme || (window.matchMedia('(prefers-color-scheme: dark)').matches ? 'dark' : 'light');
    this.currentThemeIndex = this.themes.indexOf(this.currentTheme);
    this.setTheme(this.currentTheme);
  },
};
</script>

<style scoped>
.bottom-right-container {
  display: flex;
  justify-content: flex-end;
  align-items: flex-end;
  position: fixed;
  bottom: 30px;
  right: 30px;
}
</style>

```