

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**
(БГТУ им. В.Г. Шухова)

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №2
по дисциплине «Технологии WEB-программирования»
тема: «Работа с VUE»

Выполнил: ст. группы ПВ-41
Саяпин Д.Н.
Проверил:
Картамышев С. В.

Белгород 2021 г.

Цель работы: познакомиться с основами разработки на языке JavaScript. Познакомится с основами разработки frontend приложения, в частности с использованием клиентских фреймворков (Vue js). Научиться переносить вёрстку в отдельные компоненты.

Выполнение

Vue CLI — полноценная система для быстрой разработки на Vue.js

- **CLI** (@vue/cli) — это npm-пакет, устанавливаемый глобально и предоставляющий команду vue в терминале. Он позволяет быстро создать новый проект командой vue create, или мгновенно прототипировать ваши новые идеи через vue serve. Также можно управлять проектами в графическом интерфейсе через vue ui. Мы рассмотрим, что он может делать, в следующих разделах руководства.
- **Vuex** — паттерн управления состоянием + библиотека для приложений на Vue.js. Он служит централизованным хранилищем данных для всех компонентов приложения с правилами, гарантирующими, что состояние может быть изменено только предсказуемым образом. Vuex интегрируется с официальным расширением vue-devtools (opens new window), предоставляя «из коробки» такие продвинутые возможности, как «машину времени» для отладки и экспорт/импорт слепков состояния данных.

Для начала работы на Vue необходимо выполнить следующий список действий:

1. Создать проект через консоль

```
vue create MY_project
```

2. Выбрать нужную сборку

3. Установить зависимости проекта с различными библиотеками

```
npm i axios —save
```

- vuex (Единое централ. хранилище)
- axios (Для получения и обработки данных через API)
- node-sass/sass-loader (Для работы с препроцессорами sass)

4. Установить фейковый api

```
npm install -g json-server
```

5. Добавляем db.json

```
npx json-server db.json
```

6. Далее создаем компоненты и формируем иерархию проекта

7. Получаем

```
url json-server --watch db.json
```

8. Запускаем проект

```
npm run serve
```

Код программы

Authorization.vue

```
<template>
  <section class = "authorization_vue">
    <div class="login-box">
      <div class="row">
        
      </div>
      <h2>{{this.info}}</h2>
      <form>
        <div class="user-box">
          <input type="text" v-model="username" name="" required="" class="username"/>
          <label><i class="Username"></i>
            Username
          </label>
        </div>
        <div class="user-box">
          <input type="password" v-model="password" name="" required=""
class="password"/>
          <label><i class="Unlock"></i>
            Password
          </label>
        </div>
      </form>
      <button type="submit" class = "submit-button" @click = "authorization()"
>Enter</button>
    </div>
  </section>
</template>

<script>
export default {

  name: "Authorization",
  beforeCreate: function() {
    document.body.className = 'auth';
  },
  data() {
    return {
      userId: null,
      username: null,
      password: null,
      info: 'Enter the Merch Store',
      correct: false
    }
  },
  methods: {
    authorization(){
      //тут добавление в бд корзины пользователя сделать
```

```
const params = {
  username: this.username, password: this.password
}
this.$http.post('/user', params)
.then(response => this.checkCorrect(response.data.userId))

},
checkCorrect(data){
  if(data !== -1)
  {
    this.userId = data
    this.correct = true
    this.$router.push('MainPage')
    this.info = "Success"
  }
  else
  {
    this.info = "Invalid login or password"
  }
  console.log(this.correct)
}
},
}
</script>

<style scoped src="../../assets/styles/style.css">
</style>
```

ContentPage.vue

```
<template>
<section class="content">
<Header></Header>
  <main class = "container" v-if = "product">
    <div class="product-title">
      <h1>{{ product.title }}</h1>
    </div>

    <div class = "info">
      <div class="product-image">
        
      </div>

      <div class = "product-description">
        <div class="authors">Author: {{ product.author }}</div>
        <div class="type">Type: {{ product.type }}</div>

        <div class = "buying">
          <div class = "buying-price">
            <div class="buying-pricenew-text">Price: </div>
            <div class="buying-pricenew-val">
              <span class="buying-price-val-number">{{product.price}}</span>
              <span class="buying-pricenew-val-currency">p.</span>
            </div>
          </div>
          <button class="btn" @click = "addToCart(product)"><span class="text">Add
<span>to cart</span></span></button>

          </div>
        </div>
        <hr></hr>

        <div class = "product-about">
          <h4> {{product.author}}: "{{ product.title }}"</h4>
          <div class = "annotation">
            <p>
              {{product.description}}
            </p>
          </div>
        </div>
      </div>
    </main>
    <Footer></Footer>
  </section>
</template>

<script>
import Footer from "../Footer";
import Header from "../Header";
```

```
import Product from "../Product";

export default {
  name: "ContentPage",
  components: {
    Footer,
    Header,
    Product
  },
  methods: {
    addToCart(product){
      //<добавить в бд корзины пользователя>
      const params = {
        id: product.id, count: 1
      }
      this.$http.post('/cart', params)
      //.then(response => product.inCart = response.data.inCart)
    }
  },
  data() {
    return {
      product: null
    }
  },
  created() {
    const id = this.$route.params.id
    this.$http.get('/products/' + id)
      .then(response => this.product = response.data)
  }
}
</script>

<style scoped src="../../assets/styles/content-style.css">

</style>
```

Footer.vue

```
<template>
  <footer class="footer">
    <div class="footer-text">
      ©2022 EvilPasta
    </div>

    <div class="footer-content-right">

      <a href="https://www.instagram.com/evil_pasta/"></a>

    </div>
  </footer>
</template>

<script>
export default {
  name: "Footer"
}
</script>

<style scoped src="../../assets/styles/footer_style.css">
</style>
```

Header.vue

```
<template>
  <div id="wrap">
    <header class = main-header>

      <img src = "../assets/images/logo.png" alt ="Logo" />

      <div class="header-text">
        <h1>PASTA MERCH</h1>
      </div>

      <form class="search" onsubmit="return false;">
        <input v-model="searchInfo" type="search" name="" placeholder="search"
class="input" />
        <input type="submit" name="" value="" class="submit" @click = "search()" />
      </form>

      <nav>
        <a href="#"><h2 class = "about-header">Cabinet</h2></a>
        <a href="#"><h2 class = "about-header">My cart</h2></a>
        <router-link :to="{name: 'About'}">
          <h2 class = "about-header">About</h2>
        </router-link>
      </nav>

    </header>
  </div>
</template>

<script>

export default {
  name: "Header",
  data() {
    return{
      searchInfo: null
    }
  },
  methods: {
    search(){
      this.$emit("searchChanged", this.searchInfo)
      console.log(this.searchInfo)
    }
  }
}
</script>

<style src="../assets/styles/header_style.css">
</style>
```


MainPage.vue

```
<template>
  <section>
    <Header v-on:searchChanged = "updateSearch"></Header>
    <main class="container">
      <div>
        <h1>
          Artwork, made by EvilPasta
        </h1>
      </div>
      <section class="products">
        <div class="v-for" v-for="product in products">
          <Product :product="product"></Product>
        </div>
      </section>
    </main>
    <Footer></Footer>
  </section>
</template>

<script>
import Product from "./Product";
import Header from "./Header";
import Footer from "./Footer";
import {forEach} from "core-js/internals/array-iteration";

export default {
  name: "MainPage",
  components: {
    Footer,
    Header,
    Product
  },
  beforeCreate: function() {
    document.body.className = 'main';
  },
  data(){
    return{
      products: [],
      searched: "",
      allProducts: []
    }
  },
  created() {
    this.$http.get('/products')
      .then(response => this.searchFilter(response.data))
  },
  methods: {
    searchFilter(data){
      this.allProducts = data
      this.products = []
      data.forEach((element) => {
        if (element.title.includes(this.searched)) {
```

```
                this.products.push(element)
            }
        })
    },

    updateSearch(e){
        this.searched = e
        this.searchFilter(this.allProducts)
    }
}
</script>

<style scoped src="../../assets/styles/main_style.css">

</style>
```

Product.vue

```
<template>
  <section class="product-item">
    <div>
      
    </div>
    <h4>
      <div class = "title">
        <router-link :to="{name: 'Content', params: {id: product.id}}">
          {{ product.title }}
        </router-link>
      </div>

      <div class = "author">
        {{product.author}}
      </div>
    </h4>
    <p>
      {{product.description}}
    </p>
    <div>
      <span class="price">{{ product.price }} ₺</span>
    </div>
  </section>
</template>

<script>
export default {
  name: "Product",
  props: ['product']
}
</script>

<style scoped src="../assets/styles/product_style.css">

</style>
```

Home.vue

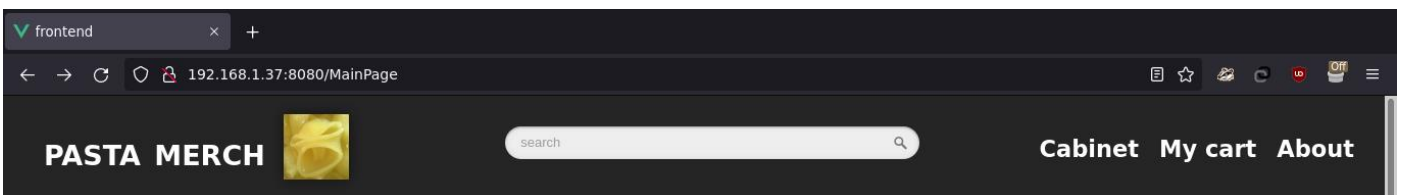
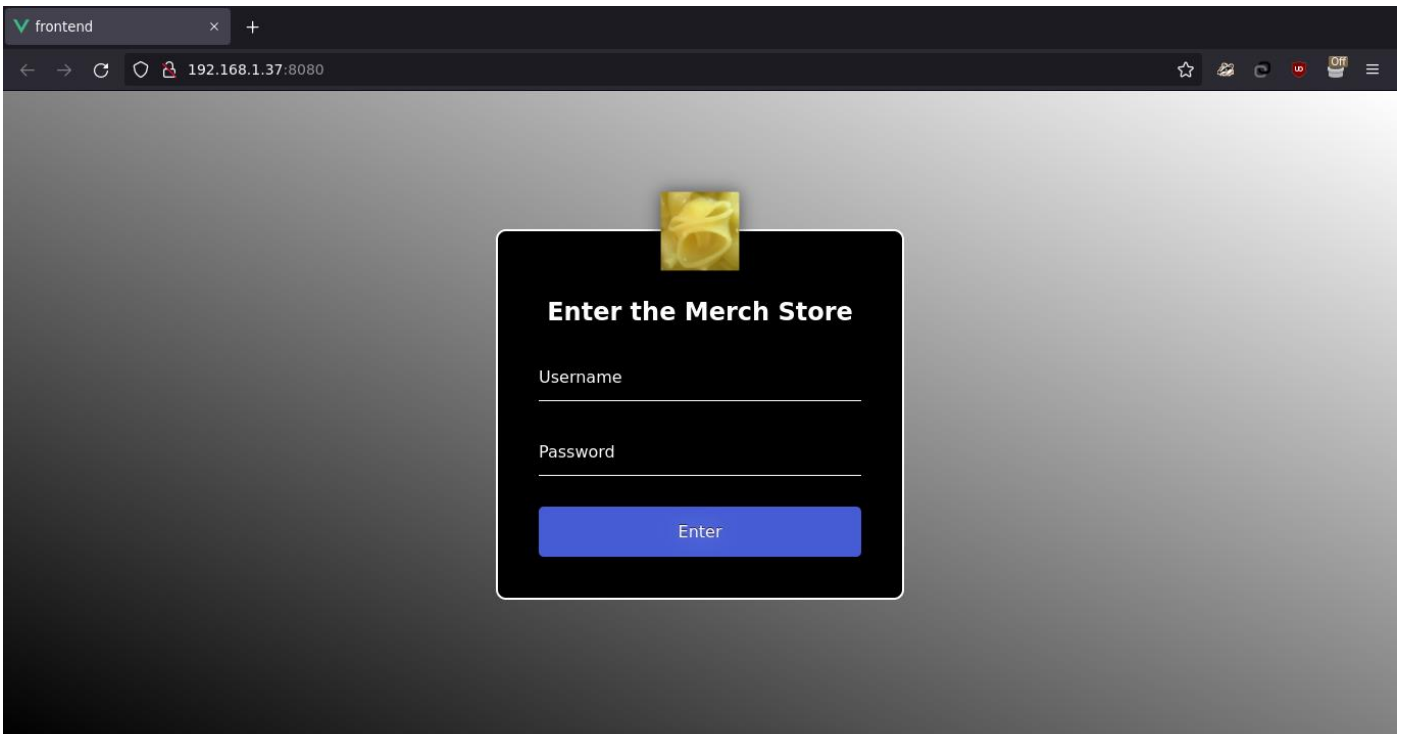
```
<template>
  <div class="home">
    <Authorization></Authorization>
  </div>
</template>

<script>

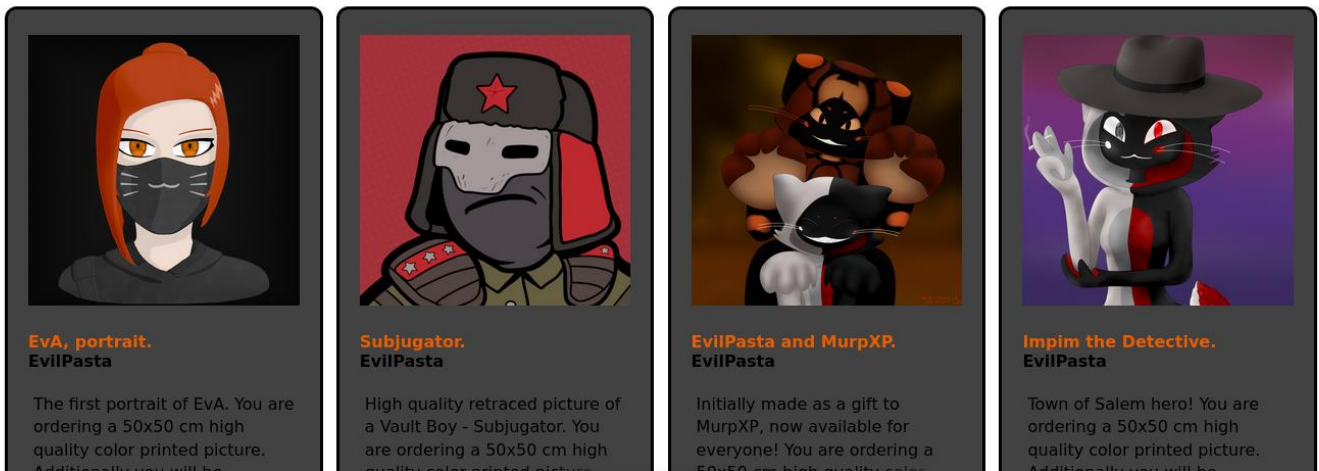
// @ is an alias to /src
import Header from '@components/Header.vue'
import Footer from "../components/Footer";
import MainPage from "../components/MainPage";
import Authorization from "../components/Authorization";

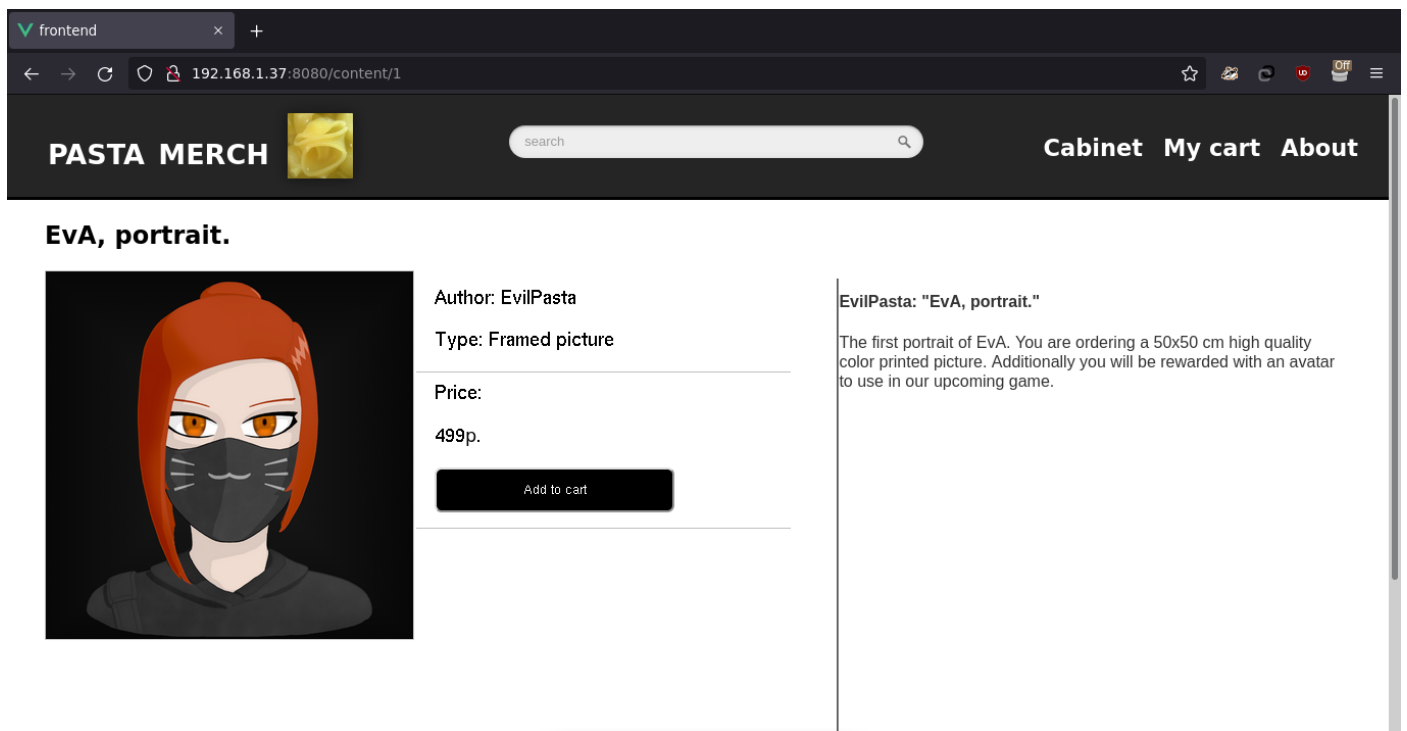
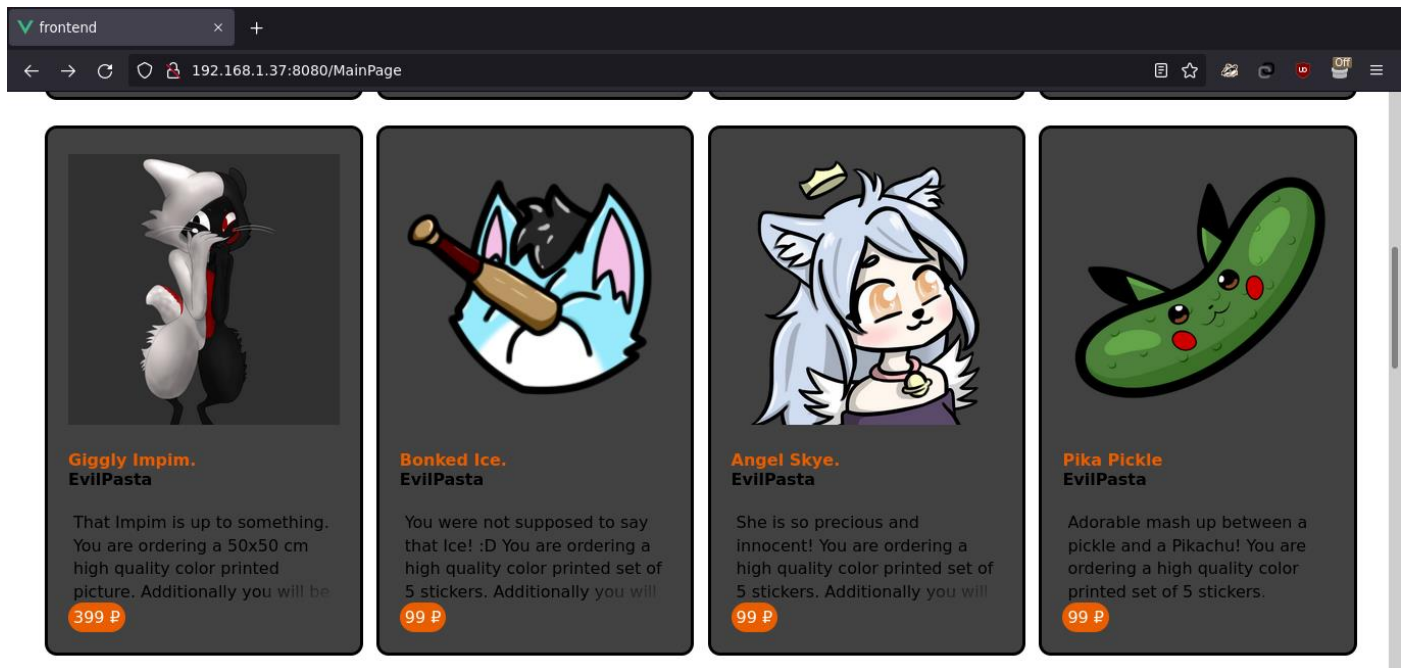
export default {
  name: 'Home',
  components: {
    Authorization,
    MainPage,
    Footer,
    Header
  }
}
</script>
```

Результат выполнения программы



Artwork, made by EvilPasta





Github: <https://github.com/EvilPasta/WebLab>