

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 №6
по дисциплине «Технологии WEB-программирования»
тема: «Работа с HTTP запросами»

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

Белгород 2021 г.

Цель работы: изучить принципы работы отправки Ajax запросов. Изучить работу отправки синхронных и асинхронных запросов в Vue JS. Получить навыки работы с промисами.

Задание к лабораторной работе:

1. Изучить возможности Vue js для отправки http запросов.
2. Выбрать подходящую библиотеку для работы с запросами.
3. Реализовать взаимодействие фронтенда с REST API, спроектированном в прошлой лабораторной работе.
4. Продемонстрировать работу взаимодействия фронтенд приложения с REST API.

Код взаимодействия фронтенд приложения с REST API

api.php

```
<?php

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;

/*
|-----
| API Routes
|-----
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| is assigned the "api" middleware group. Enjoy building your API!
|
*/

Route::middleware('auth:sanctum')->get('/user', function (Request $request) {
    return $request->user();
});

Route::prefix('products')->group(function(){
    Route::get('/', [\App\Http\Controllers\ProductController::class, 'list']);
    Route::get('/{id}', [\App\Http\Controllers\ProductController::class, 'info']);
});

Route::prefix('cart')->group(function(){
    Route::get('/{userId}', [\App\Http\Controllers\CartController::class, 'info']);
    Route::post('/', [\App\Http\Controllers\CartController::class, 'addToCart']);
});

Route::prefix('user')->group(function(){
    Route::post('/', [\App\Http\Controllers\UserController::class, 'authorization']);
});
```

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
        </div>
      </form>
    </div>
  </section>
</template>
```

```

        </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>

```

// Вход по токену

```

<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 = {
        name: 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"
        localStorage.userId = this.userId;
      }
      else

```

```

        {
            this.info = "Invalid login or password"
        }
        console.log(this.correct)
    }
},
mounted(){
    if (localStorage.userId) {
        this.userId = localStorage.userId;
    }
}
}
</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, userId: localStorage.userId
            }
            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>

```

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)) {
```

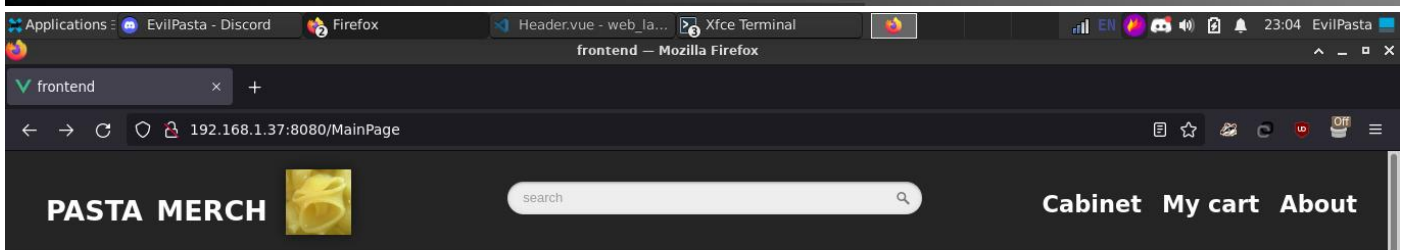
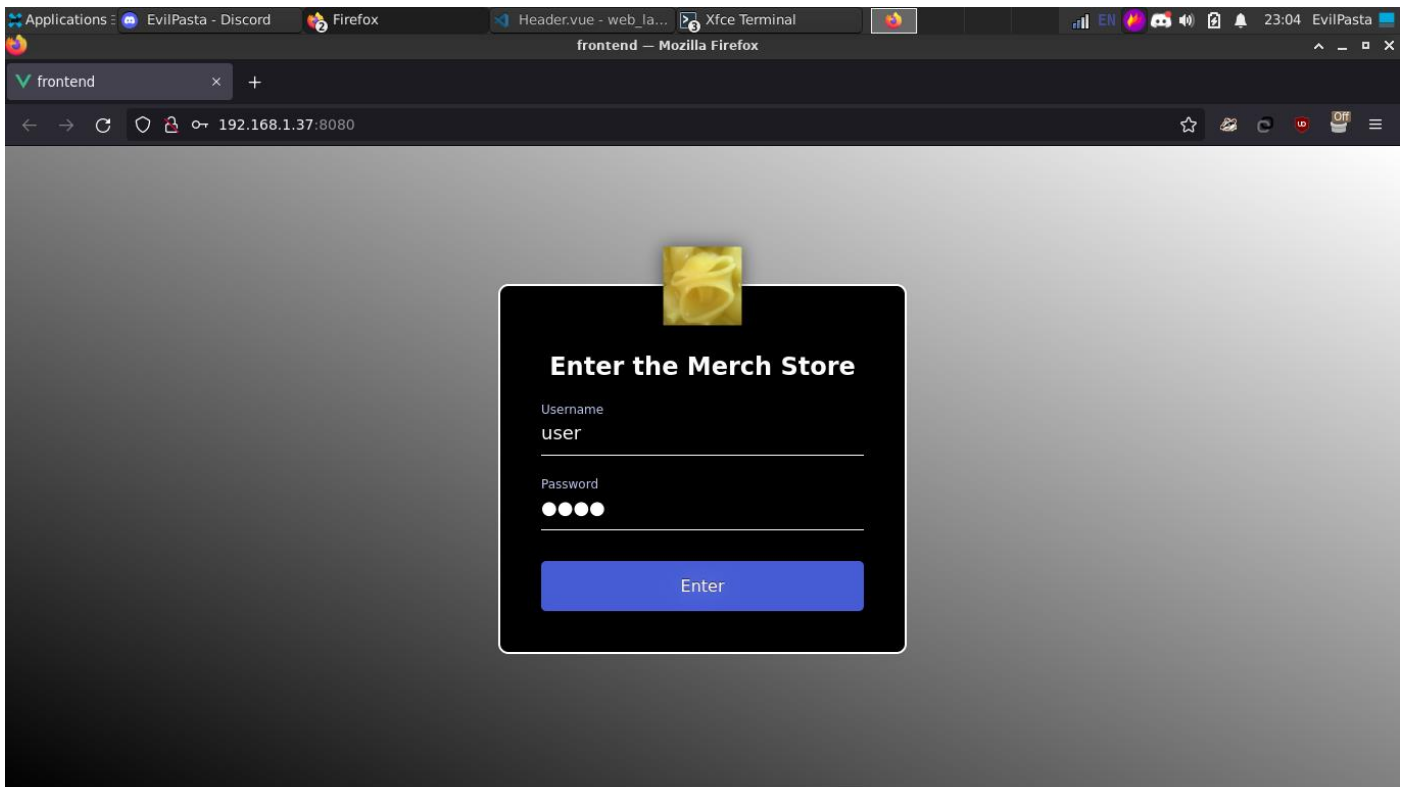
```
                this.products.push(element)
            }
        })
    },

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





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

</style>
```


Результат работы приложения




Artwork, made by EvilPasta

 <p>EvA, portrait. EvilPasta</p> <p>The first portrait of EvA. You are ordering a 50x50 cm high quality color printed picture. Additionally you will be</p>	 <p>Subjugator. EvilPasta</p> <p>High quality retraced picture of a Vault Boy - Subjugator. You are ordering a 50x50 cm high quality color printed picture.</p>	 <p>EvilPasta and MurpXP. EvilPasta</p> <p>Initially made as a gift to MurpXP, now available for everyone! You are ordering a 50x50 cm high quality color</p>	 <p>Impim the Detective. EvilPasta</p> <p>Town of Salem hero! You are ordering a 50x50 cm high quality color printed picture. Additionally you will be</p>
---	---	--	--

Applications : EvilPasta - Discord Firefox Header.vue - web_la... Xfce Terminal frontend — Mozilla Firefox

frontend


192.168.1.37:8080/MainPage



Giggly Impim.
EvilPasta

That Impim is up to something. You are ordering a 50x50 cm high quality color printed picture. Additionally you will be


399 ₺



Bonked Ice.
EvilPasta

You were not supposed to say that Ice! :D You are ordering a high quality color printed set of 5 stickers. Additionally you will

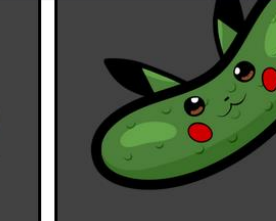
99 ₺



Angel Skye.
EvilPasta

She is so precious and innocent! You are ordering a high quality color printed set of 5 stickers. Additionally you will


99 ₺





Pika Pickle
EvilPasta


Adorable mash up between a pickle and a Pikachu! You are ordering a high quality color printed set of 5 stickers.

99 ₺










Applications : EvilPasta - Discord Firefox Header.vue - web_la... Xfce Terminal frontend — Mozilla Firefox

frontend


192.168.1.37:8080/MainPage



Tiny boi Black
EvilPasta

Adorable Tiny Boi of your preferred color! You are ordering a high quality color printed set of 5 stickers.


99 ₺



Tiny boi Blue
EvilPasta

Adorable Tiny Boi of your preferred color! You are ordering a high quality color printed set of 5 stickers.


99 ₺



Tiny boi Brown
EvilPasta

Adorable Tiny Boi of your preferred color! You are ordering a high quality color printed set of 5 stickers.


99 ₺

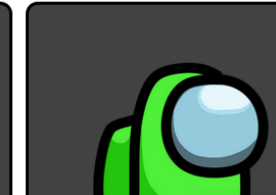



Tiny boi Cyan
EvilPasta

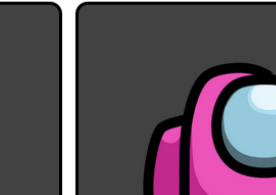
Adorable Tiny Boi of your preferred color! You are ordering a high quality color printed set of 5 stickers.

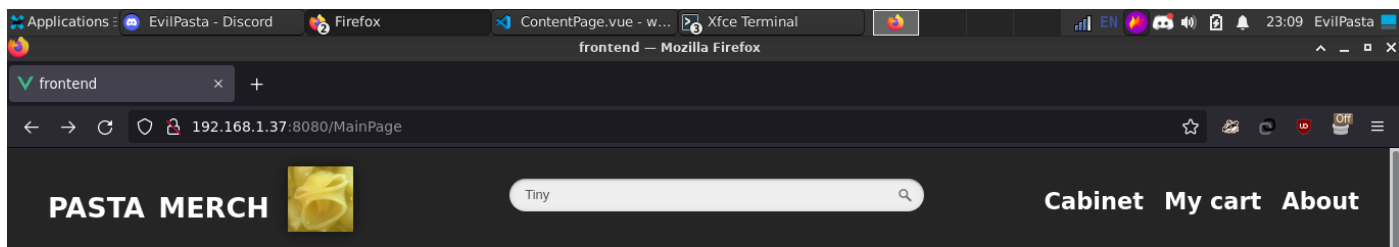
99 ₺











Artwork, made by EvilPasta



Tiny boi Black
EvilPasta

Adorable Tiny Boi of your preferred color! You are ordering a high quality color printed set of 5 stickers



Tiny boi Blue
EvilPasta

Adorable Tiny Boi of your preferred color! You are ordering a high quality color printed set of 5 stickers



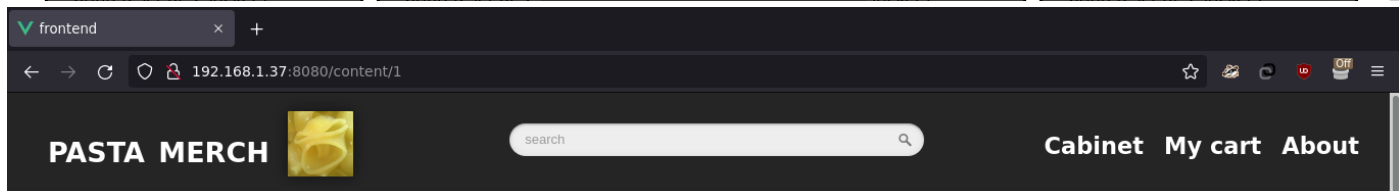
Tiny boi Brown
EvilPasta

Adorable Tiny Boi of your preferred color! You are ordering a high quality color printed set of 5 stickers



Tiny boi Cyan
EvilPasta

Adorable Tiny Boi of your preferred color! You are ordering a high quality color printed set of 5 stickers



EVA, portrait.



Author: EvilPasta

Type: Framed picture

Price:

499p.

Add to cart

EvilPasta: "EVA, portrait."

The first portrait of EvA. You are ordering a 50x50 cm high quality color printed picture. Additionally you will be rewarded with an avatar to use in our upcoming game.

Angel Skye.



Author: EvilPasta

Type: Sticker

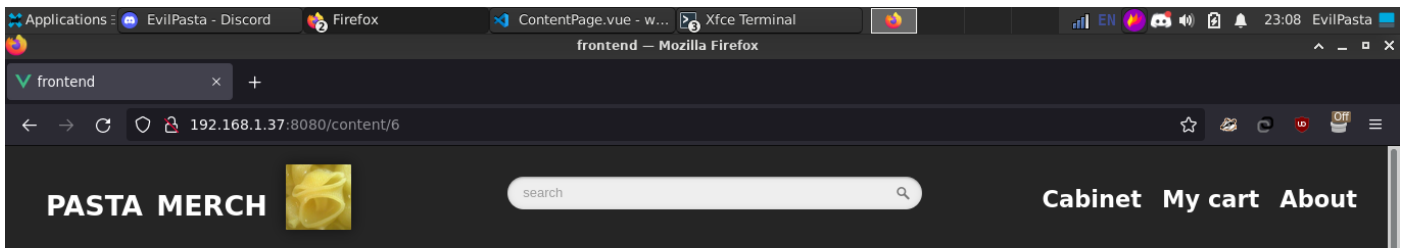
Price:

99p.

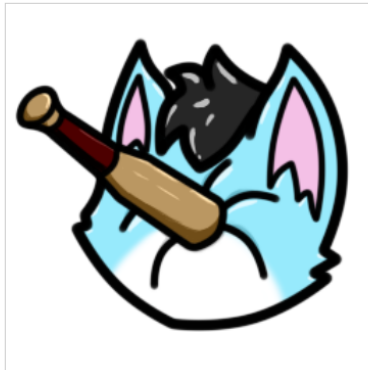
Add to cart

EvilPasta: "Angel Skye."

She is so precious and innocent! You are ordering a high quality color printed set of 5 stickers. Additionally you will be rewarded with an avatar to use in our upcoming game.



Bonked Ice.



Author: EvilPasta

Type: Sticker

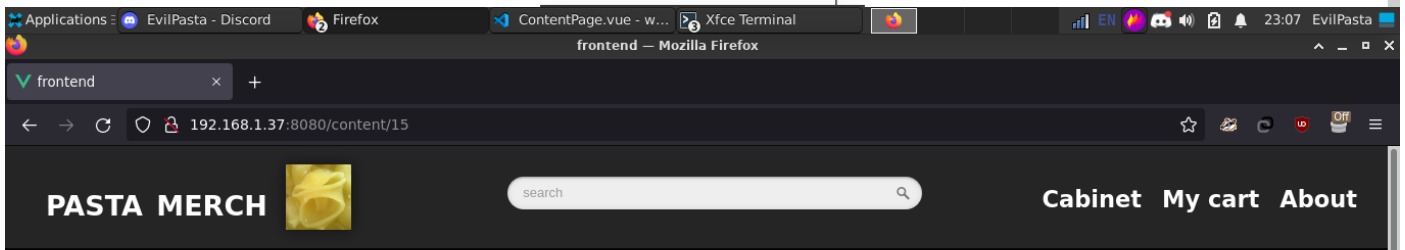
Price:

99p.

Add to cart

EvilPasta: "Bonked Ice."

You were not supposed to say that Ice! :D You are ordering a high quality color printed set of 5 stickers. Additionally you will be rewarded with an avatar to use in our upcoming game.



Tiny boi Orange



Author: EvilPasta

Type: Sticker

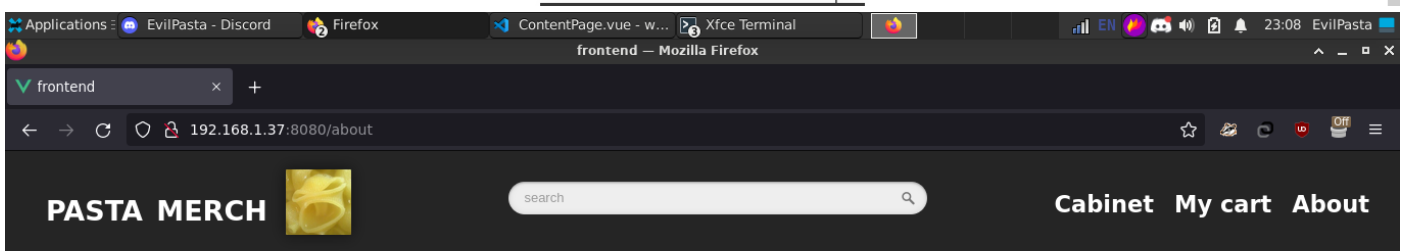
Price:

99p.

Add to cart

EvilPasta: "Tiny boi Orange"

Adorable Tiny Boi of your preferred color! You are ordering a high quality color printed set of 5 stickers. Additionally you will be rewarded with an avatar to use in our upcoming game.



About EvilPasta

I am an artist, a game developer and a streamer! Thank you for all your support, without you we would never have a community of our size, and we would not have spent so much time having fun together!

©2022 EvilPasta



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