

Национальный исследовательский университет ИТМО



Лабораторная работа №3
«Разработка одностраничного веб-приложения (SPA) с использованием
фреймворка Vue.JS»

По дисциплине
«Фронт-энд разработка»

Выполнил:
Кривцов П.А.
Группа:
К33402
Преподаватель:
Добряков Д.И.

Санкт-Петербург

2023 г
ЗАДАНИЕ

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

ВЫПОЛНЕНИЕ

Исходный код: <https://github.com/upaffy/ITMO-ICT-Frontend-2022/tree/main/labs/K33402/Krivtsov%20Pavel/Lab3>

В процессе работы были реализованы следующие представления:

views	
✓ ChangePasswordView.vue	Страница смены пароля
✓ CurrencyView.vue	Страница информации о валюте
✓ LoginView.vue	Страница авторизации
✓ MarketView.vue	Страница с отображением всех валют
✓ ProfileView.vue	Профиль с валютой пользователя
✓ RegisterView.vue	Страница регистрации
✓ WelcomeView.vue	Начальная страница

И следующие компоненты:

components	
✓ BuySellButtonsBlock.vue	Кнопки “купить” и “продать” для валюты
✓ ChangePasswordForm.vue	Форма для смены пароля
✓ CurrenciesList.vue	Компонент для отображения списка валют
✓ CurrencyChart.vue	Компонент для отображения графика цен
✓ CurrencyFiltersBlock.vue	Блок с кнопками сортировки и поля для поиска
✓ FooterBlock.vue	Футер
✓ HeaderNavBar.vue	Хэдер - ссылки и динамическая кнопка profile
✓ LoginForm.vue	Форма для авторизации
✓ ProfileUserBlock.vue	Блок кнопок “сменить пароль” и “выйти”
✓ RegisterForm.vue	Форма регистрации
✓ UserCurrenciesList.vue	Отображение списка валют пользователя
✓ WelcomeSection.vue	Инструкции по использованию сайта

Для представлений были созданы лэйауты:

- Default layout – для обычных страниц

```
<template>
  <HeaderNavBar/>
  <main class="min-vh-100">
    <slot/>
  </main>
  <FooterBlock/>
</template>

<script>
import HeaderNavBar from "../components/HeaderNavBar.vue";
import FooterBlock from "../components/FooterBlock.vue";

export default {
  name: "DefaultLayout",
  components: {
    HeaderNavBar,
    FooterBlock
  }
}
</script>
```

- AuthLayout – для страниц с полями для ввода данных о пользователе (логин, пароль, почта)

```
<template>
  <main class="text-center sign-body">
    <div class="sign-form w-100 m-auto">
      <slot/>
    </div>
  </main>
</template>

<script>
export default {
  name: "AuthLayout"
}
</script>
```

- AppLayout – конфигурирующий файл

```
<template>
  <component :is="layout">
    <slot />
  </component>
</template>

<script>
import DefaultLayout from "../DefaultLayout.vue";
import {shallowRef, watch} from 'vue'
import { useRoute } from 'vue-router'
export default {
  name: 'AppLayout',
  setup() {
    const layout = shallowRef(DefaultLayout)
    const route = useRoute()
    watch(
      () => route.meta,
      async meta => {
        try {
          const component = await
import(`../layouts/${meta.layout}.vue`)
          layout.value = component?.default || DefaultLayout
        } catch (e) {
          layout.value = DefaultLayout
        }
      },
      { immediate: true }
    )
    return { layout }
  }
}
</script>
```

Содержание роутера:

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

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

```

routes: [
  {
    path: '/',
    name: 'welcome',
    component: () => import('@views/WelcomeView.vue'),
    meta: {
      layout: 'DefaultLayout'
    }
  },
  {
    path: '/login',
    name: 'login',
    component: () => import('@views/LoginView.vue'),
    meta: {
      layout: 'AuthLayout'
    }
  },
  {
    path: '/register',
    name: 'register',
    component: () => import('@views/RegisterView.vue'),
    meta: {
      layout: 'AuthLayout'
    }
  },
  {
    path: '/currency/:id',
    name: 'currency',
    component: () => import('@views/CurrencyView.vue'),
    meta: {
      layout: 'DefaultLayout'
    }
  },
  {
    path: '/market',
    name: 'market',
    component: () => import('@views/MarketView.vue'),
    meta: {
      layout: 'DefaultLayout'
    }
  },
  {
    path: '/profile',
    name: 'profile',
    component: () => import('@views/ProfileView.vue'),
    meta: {
      layout: 'DefaultLayout'
    }
  },
  {
    path: '/change_password',
    name: 'change_password',
    component: () => import('@views/ChangePasswordView.vue'),
    meta: {
      layout: 'AuthLayout'
    }
  }
]
}))

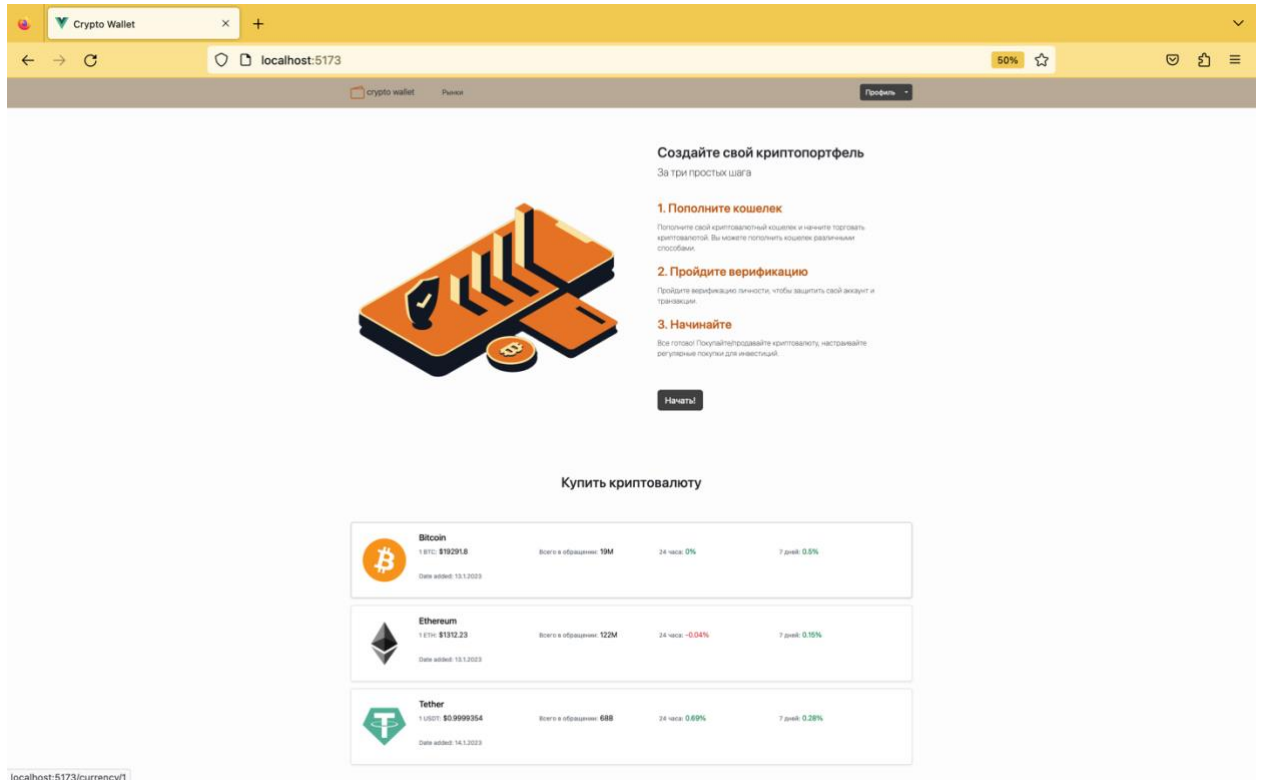
export default router

```

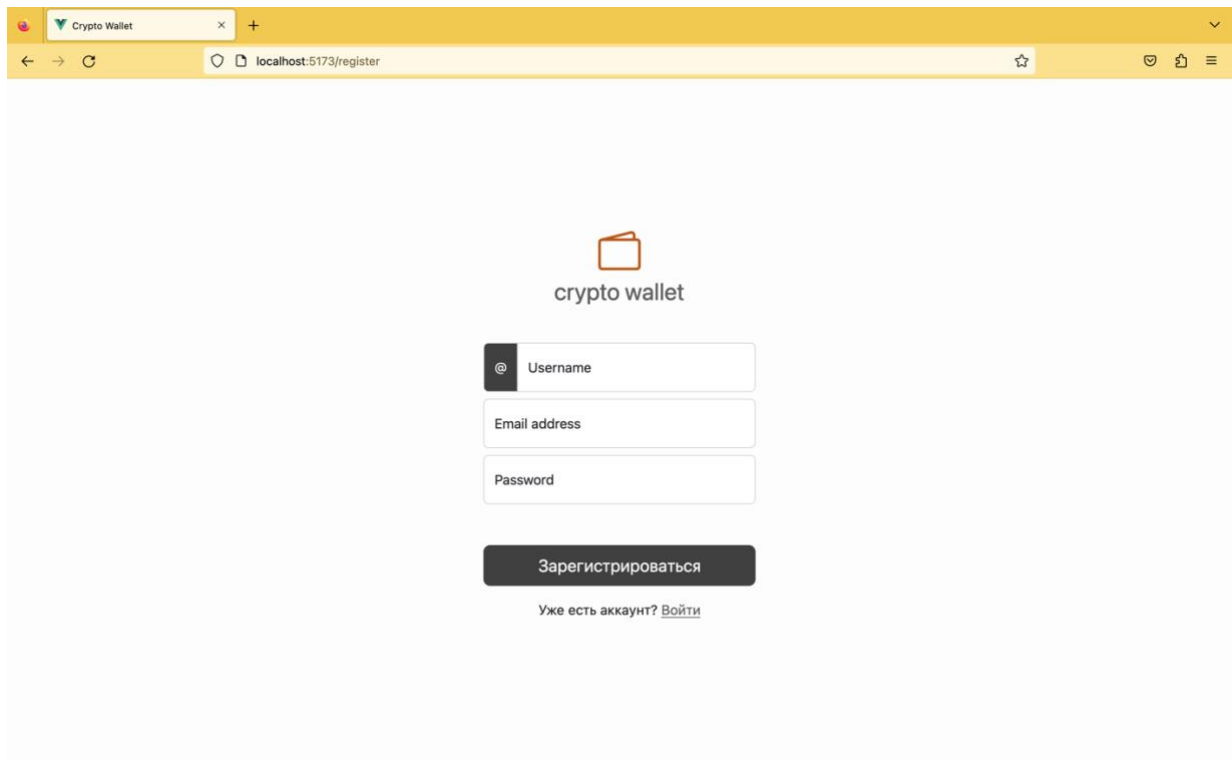
Внешний вид страниц:

1. Начальная (приветственная) страница

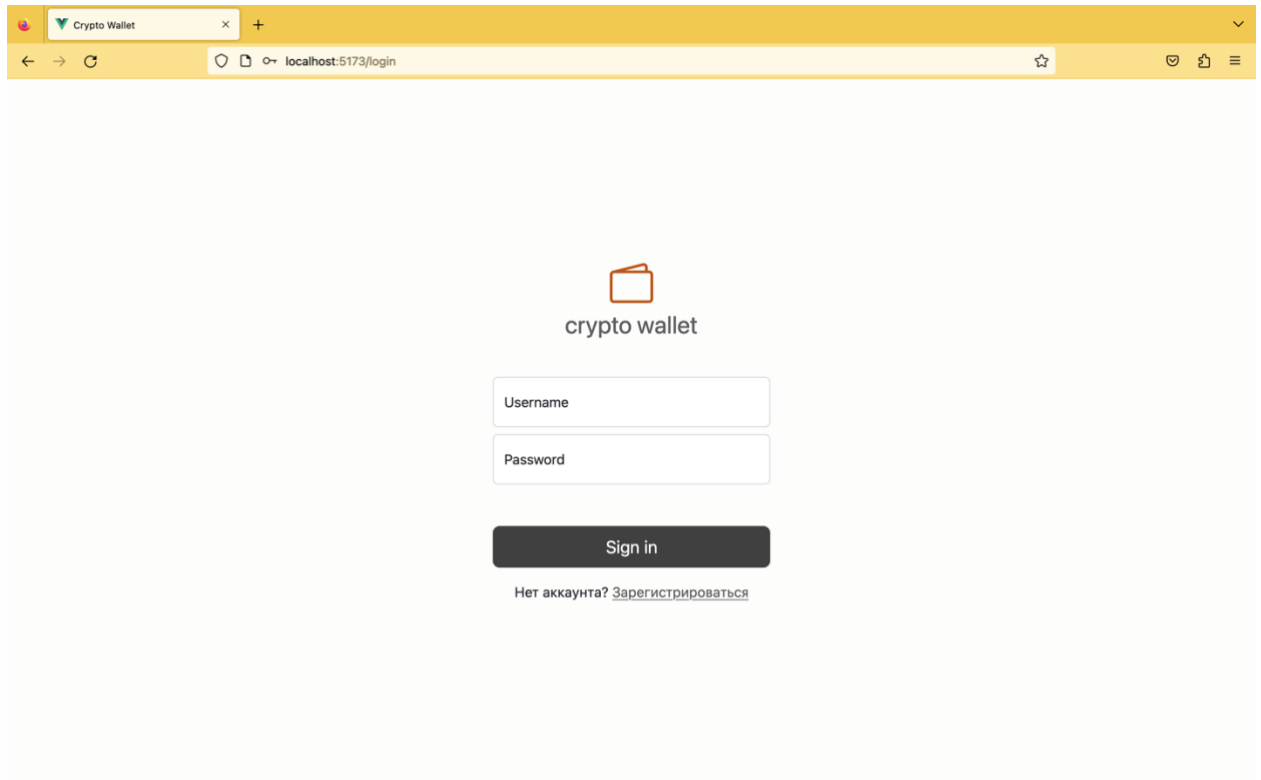
Содержит компоненты WelcomeSection и CurrenciesList, а так же футер и хедер, входящие в лэйаут



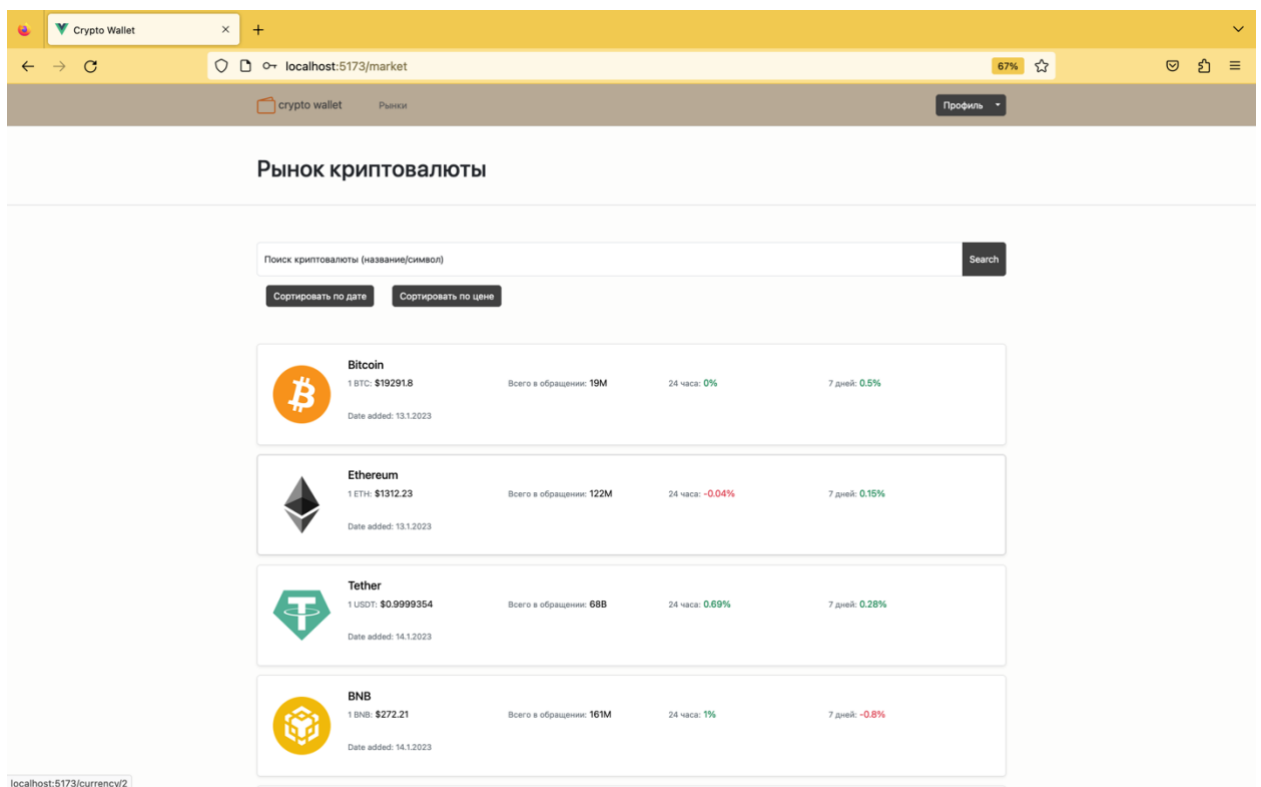
2. Страница регистрации – содержит компонент RegisterForm



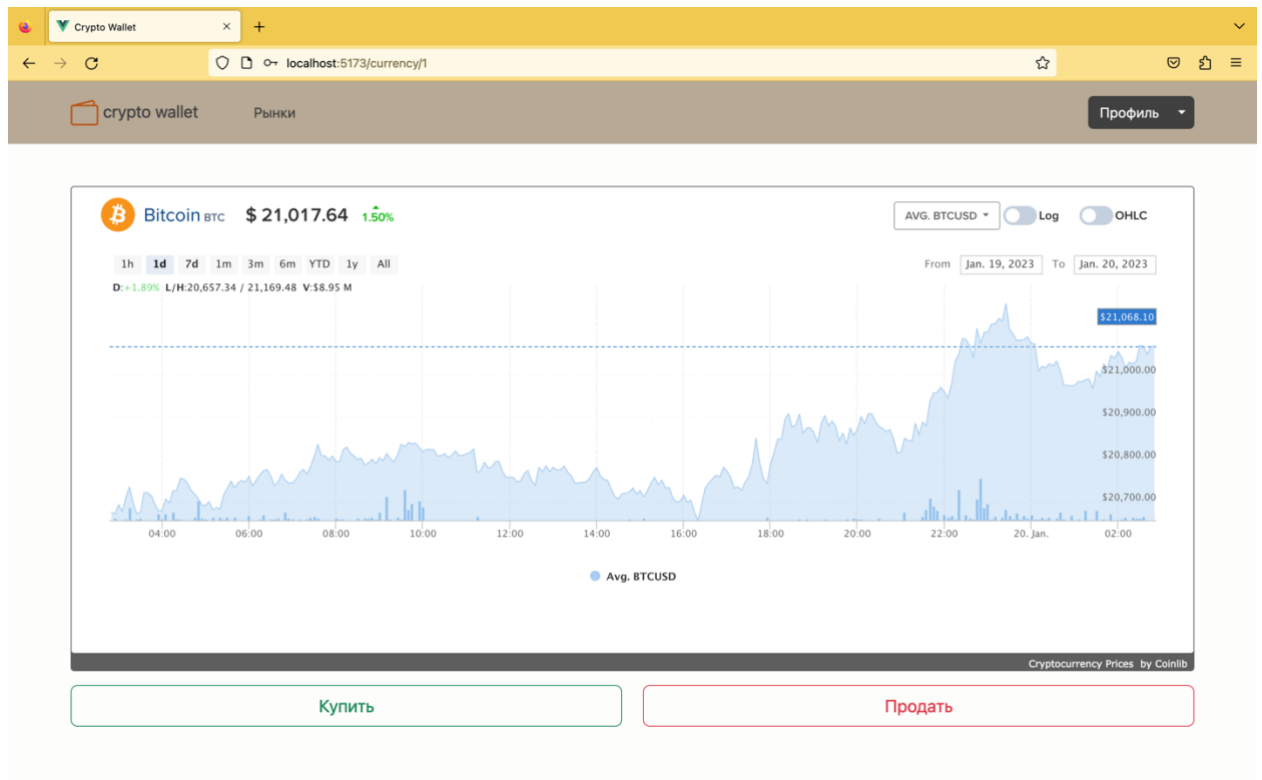
3. Страница авторизации – содержит компонент LoginForm



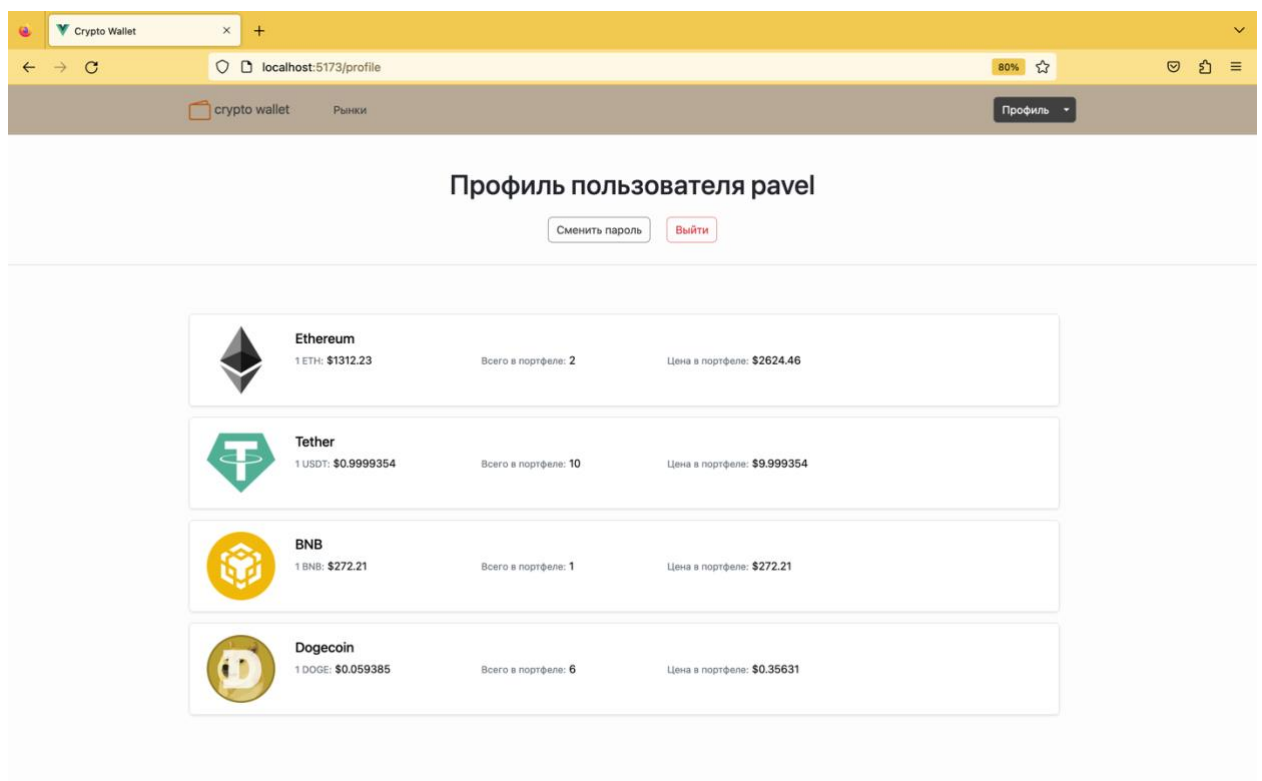
4. Рынок криптовалют – содержит компоненты CurrencyFiltersBlock и CurrenciesList



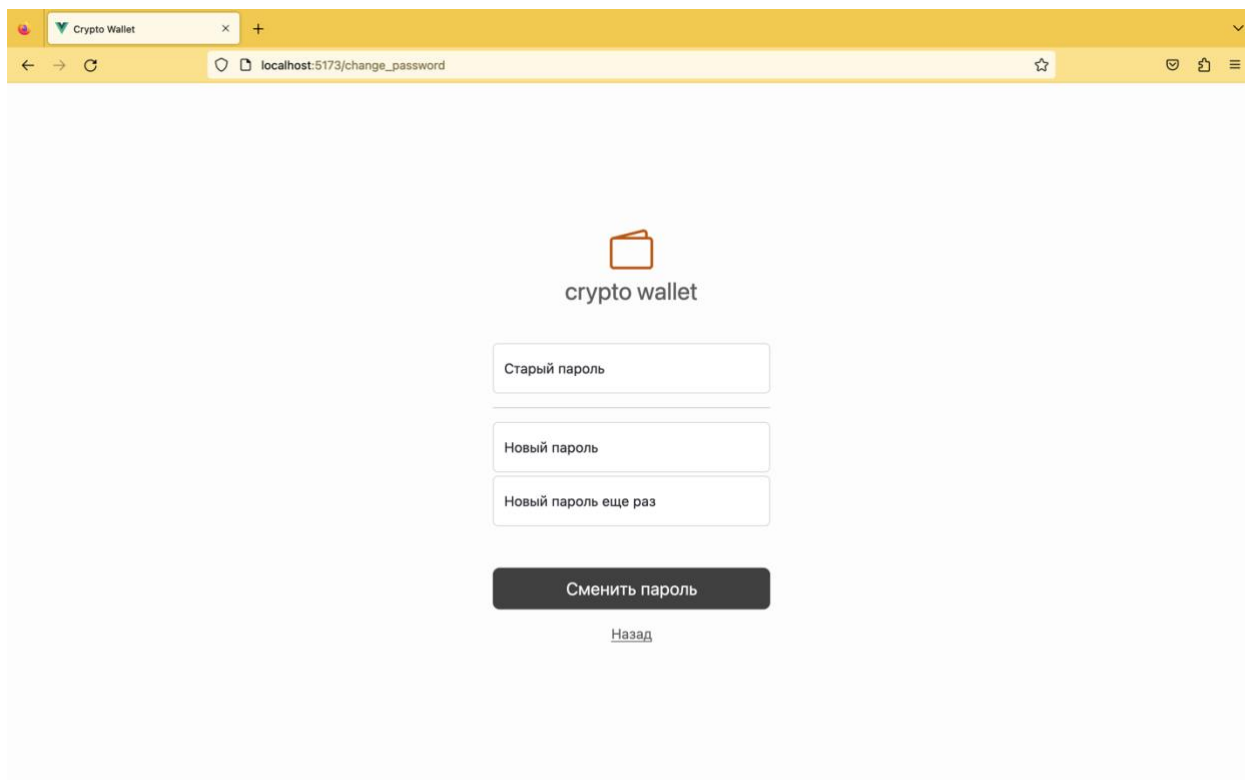
5. Информация о криптовалюте – содержит компоненты CurrencyChart и BuySellButtonsBlock



6. Профиль пользователя – содержит компоненты ProfileUserBlock и UserCurrenciesList



7. Страница смены пароля – содержит компонент ChangePasswordForm



The screenshot shows a web browser window with the title "Crypto Wallet". The address bar displays "localhost:5173/change_password". The page content is centered and features the "crypto wallet" logo, which consists of an orange wallet icon and the text "crypto wallet". Below the logo are three input fields for password entry, labeled "Старый пароль", "Новый пароль", and "Новый пароль еще раз". At the bottom of the form is a dark button labeled "Сменить пароль" and a link labeled "Назад".

Crypto Wallet

localhost:5173/change_password

crypto wallet

Старый пароль

Новый пароль

Новый пароль еще раз

Сменить пароль

[Назад](#)

ВЫВОДЫ

В процессе работы была реализована клиентская часть криптобиржи. Я познакомился с фреймворком vue.js, изучил его функционал. Реактивный подход делает этот фреймворк сложнее для понимания, но от того интереснее его изучение.