

**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: Пономарев
Константин
Группа К33402

Проверил:
Добряков Д. И.

Санкт-Петербург

2023 г.

Задача

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
- Использование composable

Ход работы

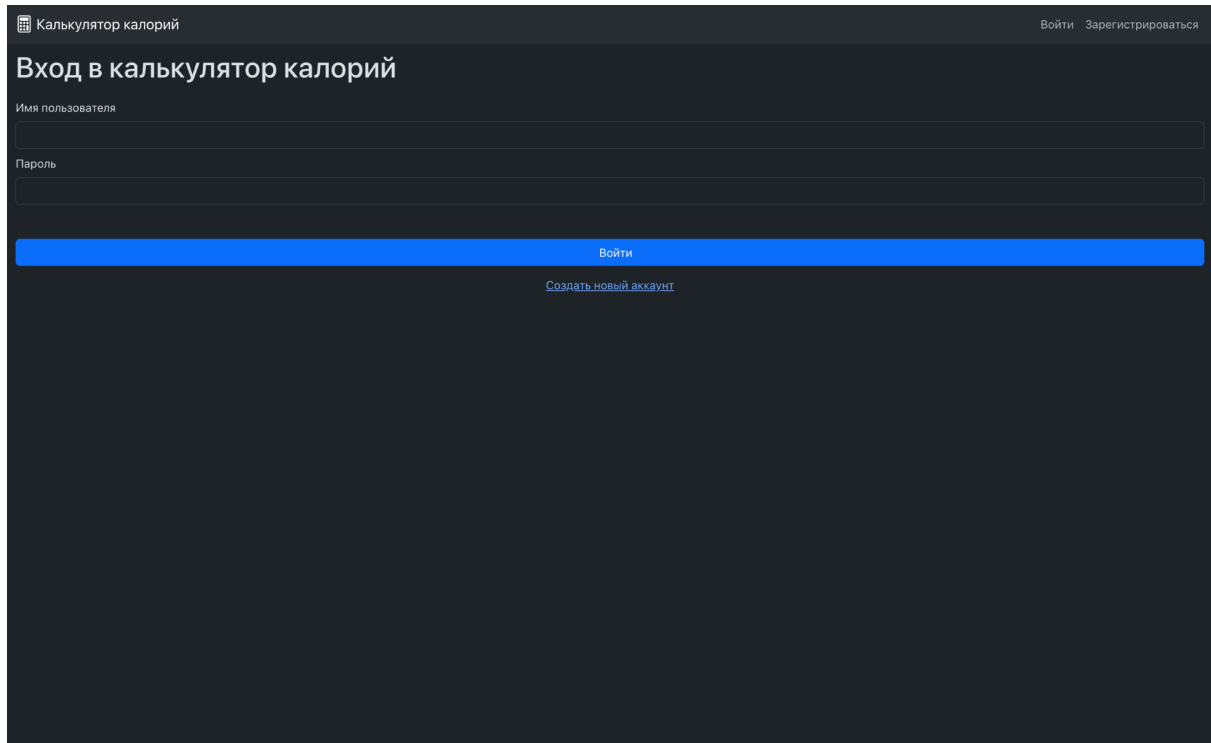
Прежде всего в этой лабораторной работе мне хотелось попробовать все и сразу, начиная от использования встроенных вещей во vue, заканчивая pinia и axios и др. библиотеками, поэтому первым делом я решил подумать над архитектурой. Мне очень нравится подход: “тонкий клиент - толстый сервер”, который заключается в том, что приложение, неважно на какой платформе (android, ios или web), должно грубо говоря показывать то, что отдает сервер, проводить какие-то базовые операции по валидации, какой-то обработке и отправке, поэтому я решил делать все таким образом. Но поскольку у меня нет бэка, мне пришлось заиспользовать json-server, который позволяет мокировать запросы, однако он тоже особо не гибкий, поэтому пришлось костылять и прописывать некоторые значения в коде.

После этого я стал думать над функциональностью сайта, исходя из минимальных требований, получилось, что мой сайт имеет несколько страниц:

- Вход
- Регистрация
- Параметры тела
- Главная, которая содержит основную информацию о вычисленных значениях калорий, белков, углеводов, жиров
- Поиск еды, в котором можно найти продукт, который был съеден и добавить его для перерасчета дневного потребления

Далее рассмотрим каждую страницу по отдельности, я постараюсь расписать основной функционал:

Страница входа

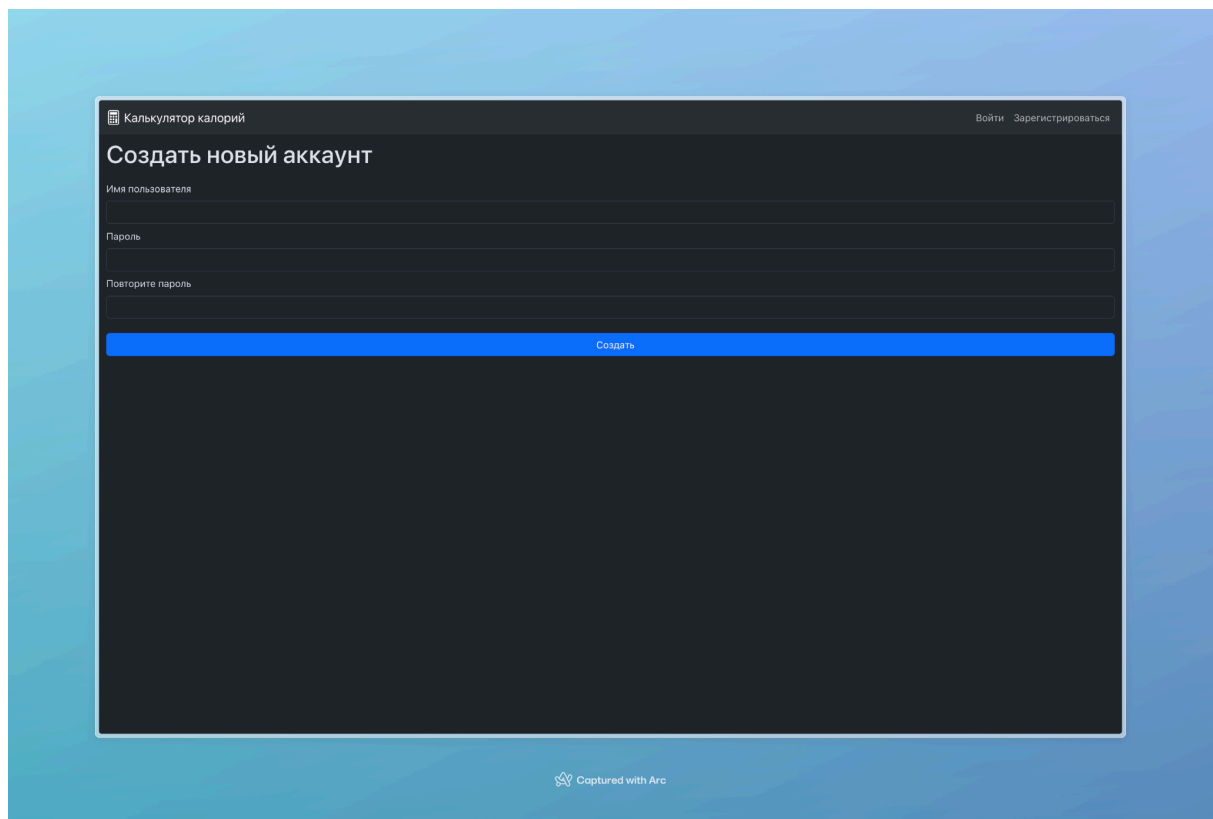


The screenshot shows a dark-themed login page for a "Калькулятор калорий" (Calorie Calculator). At the top left is the site name with a small icon, and at the top right are links for "Войти" (Login) and "Зарегистрироваться" (Register). The main heading is "Вход в калькулятор калорий". Below it are two input fields: "Имя пользователя" (Username) and "Пароль" (Password). A prominent blue button labeled "Войти" is centered below the fields. Underneath the button is a link that says "Создать новый аккаунт" (Create new account).

Собственно страница входа является приветственной и содержит в себе поля ввода данных пользователя. Также, если пользователь еще не зарегистрирован, то он может создать новый аккаунт, нажав на соответствующую кнопку, которая редиректит на страницу регистрации.

Из интересного здесь особо ничего нет, просто минимальная валидация полей, к-ая требует хоть какого-то ввода символа. Перейдем к следующей странице: Регистрация

Регистрация



Регистрация была поинтереснее, поскольку именно тут я тренировался и вообще смотрел возможности, которые предоставляет bootstrap и vue

Собственно поля абсолютно такие же как и на входе, но с кастомной валидацией, которая выглядит вот так:

```
usernameError() {  
  
  const usernamePattern = /^[a-zA-Z0-9_]+$/;  
  
  if (!this.form.username) {  
  
    return 'Введите имя';  
  
  } else if (!usernamePattern.test(this.form.username)) {  
  
    return 'Имя пользователя может содержать буквы, цифры, а также нижнее  
подчеркивание';  
  
  }  
  
  return '';  
}
```

```
}  
  
passwordMatch() {  
  
  return this.form.password === this.form.repeatPassword  
  
},
```

Из интересного, тут я решил прям заморочиться, чтобы по нажатию на кнопку появлялся лоадер на ней и менялся текст, это мне удалось добиться, добавив в data поле showLoading, которое при смене тригерило показ лоадера и подставляло другой текст. Еще я тут решил не пользоваться v-if и другими прелестями, а ограничиться v-model, поэтому в коде можно увидеть вот такие конструкции: :class="{ 'invisible': !form.showLoading }", опять же, просто ради изучения возможностей

После успешной регистрации, пользователя отправляет на экран настройки тела, а токен с авторизации сохраняется в persist store

Параметры тела

Калькулятор калорий

Поиск еды Профиль

Параметры вашего тела

Общая информация

☒ Мужчина ☐ Женщина

Рост, см: 170 Вес, кг: 70 Возраст, лет: 21

Ваша активность

Средняя

Ежедневно гуляю не меньше часа

Ваша цель

Формула расчета

☒ Харриса-Бенедикта ☐ Миффлина Сан-Жеора

Captured with Arc

На этой странице я понял, насколько круто использовать bootstrap, он спокойно позволяет делать крутые вещи, например вот такой ползунок, который при движении триггерит смену описания, а также простейшее использование radio кнопок, жаль, что такого не дает библиотека material в андроиде(

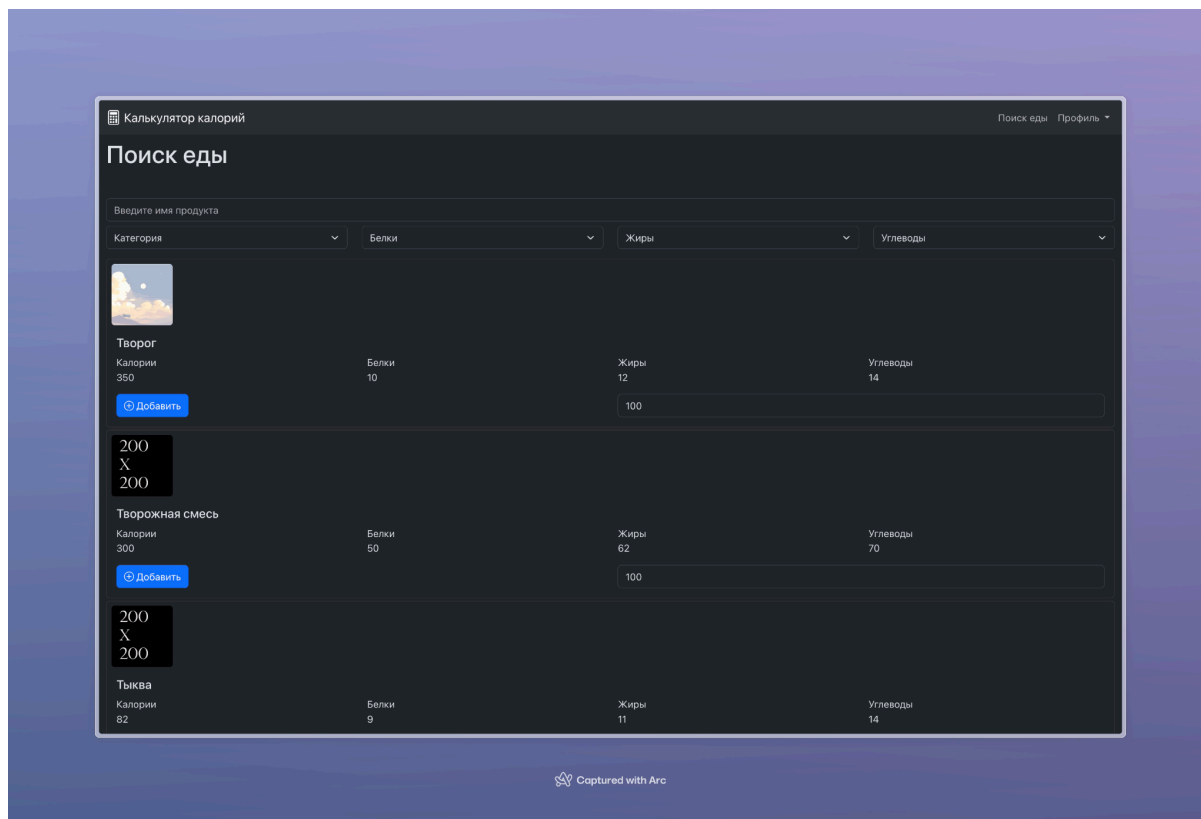
Из интересного: именно тут я узнал, что в JS нет enum'ов, очень мало синтаксического сахара, например как в kotlin блок when, а еще при создании классов нельзя указать тип, только jsdoc comment приходит на помощь. В остальном ничего особенного, везде как есть. При нажатии на кнопку, мы идем на сервер, к-ый у себя вычисляет параметры, а клиент редиректит на главную

Главная



Не обращайте внимания на иконку, она бралась с моков, а искать разную ради красоты не хотелось. Из интересного тут, я попробовал потыкать `v-if`, сделал свой кастомный компонент `FoodItem`, который переиспользовал в “Поиск еды”. При входе на эту страницу мы дергаем метод `loadData`, который идет к бэку для получения параметров по калориям, белкам, жирам, углеводам. Все прогрессы динамические, если пользователь добавил еду, сайт это красиво анимирует

Поиск еды



Наверное тоже одна из самых интересных страниц, тут у меня есть категории еды, фильтр по белкам, жирам, углеводам, но добавил я их чисто для показа того, что умею в фильтрацию, мало параметров, но в будущем, если в след. семестре мы будем сюда прикручивать бэк, я зарефакторю это дело. Давайте обратим внимание на поисковую строку. Я не стал делать кнопку “Поиск”, потому что хотел делать интерактивный ввод, т.е. введя один символ, уже начиналась фильтрация, в этом мне помогло использование `v-model + v-for`, который брал данные с `data`. Еще можно посмотреть на `FoodItem`, у которого появилось поле ввода граммовки, при том тоже дает сразу же фидбек по вычислению параметров блюда в зависимости от цифры, а также кнопка добавить с `svg` иконкой (взятой из спрайта). По нажатию на нее, мы кладем ее в стор, в котором есть вся инфа о соединенных продуктах

Вывод

После долгих лет работы в андроид, мой мозг перевернулся на 180, было довольно тяжело сначала сразу въехать в то, как все устроено в вебе, было очень много вопросов по css, но bootstrap очень сильно облегчил эту задачу тем, что почти все умеет делать сам. Мне понравился Vue, он довольно простой, его data, computed-свойства очень сильно упрощали работу, нежели если бы я работал чисто pinia. У меня все равно есть множество вопросов, но считаю, что мой первый сайт получился достойным