

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
УНИВЕРСИТЕТ ИТМО

Факультет инфокоммуникационных технологий

Образовательная программа 09.03.03

Направление подготовки (специальность) Мобильные сетевые технологии

О Т Ч Е Т

о курсовой работе

Тема задания: разработка клиентской части сервиса прогноза погоды средствами фреймворка Vue.JS

Обучающийся Матрохина Анна Сергеевна, К33402

Руководитель: Добряков Д. И., преподаватель

Оценка за курсовую работу: ____

Подписи членов комиссии:

(подпись)

(подпись)

(подпись)

Дата 25.01.2022

Санкт-Петербург
2022

ВВЕДЕНИЕ	3
Актуальность	3
Цели и задачи	3
ГЛАВА 1. СРЕДСТВА РАЗРАБОТКИ И ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ	4
1.1 Средства разработки	4
1.2 Функциональные требования	4
1.3 Страницы сервиса	4
ГЛАВА 2. Функционал сервиса	6
2.1. Визуальный интерфейс	6
2.2. Код	22
Выводы по работе	24
СПИСОК ЛИТЕРАТУРЫ	25

ВВЕДЕНИЕ

Актуальность

Сервис прогноза погоды – идеальный кейс для закрепления полученных знаний в рамках курса по фронтенд разработке. У подобных сервисов по получению данных о погоде есть множество функционала: от получение простых данных (например, количество градусов, как ощущается погода, количество осадков, скорость ветра, параметры влажности и тд) до просмотра динамических карт с изменением температуры по минутам. Я считаю, что будущее стоит за персонализированными сервисами. К сожалению, сейчас на рынке множество решений, которые содержат большое количество рекламы и избыточный функционал. Поэтому целью в рамках этой курсовой работы я поставила: разработка клиентской части сервиса прогноза погоды средствами фреймворка Vue.JS с минимально нужным функционалом и приятным дизайном.

Цели и задачи

1. Определение средств разработки
2. Определение функциональных требований
3. Создание адаптивной верстки сайта с помощью Bootstrap
4. Созданий функциональной части с помощью Vue.js, Axios и Vuex

ГЛАВА 1. СРЕДСТВА РАЗРАБОТКИ И ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

1.1 Средства разработки

Vue.js – это молодой фреймворк, который был создан в 2014 году, вторая версия вышла в 2016 году. Некоторые разработчики называют его библиотекой за малый размер и простоту использования. В рамках этой курсовой работы мы используем Vue, так как это очень удобный фреймворк из-за полной приятной документации и из-за компонентного подхода. На Vue разработаны такие сервисы, как:

- Gitlab
- Euronews
- Adobe Portfolio
- Behance
- Alibaba
- Trustpilot
- Vice
- Nintendo
- BMW

1.2 Функциональные требования

1. Vue.js 2.x версии
2. Библиотека компонентов Bootstrap-Vue
3. Адаптивная верстка
4. Корректная работа приложения

1.3 Страницы сервиса

В финальной версии сервиса для получения прогноза погоды в разных городах реализовано 8 страниц:

1. Авторизация
2. Регистрация
3. Сброс пароля
4. Сообщение об успешном сбросе пароля
5. Личный кабинет с добавленными городами
6. Главная страница с поиском городов и фильтрацией по дате

7. Информация о погоде в городе на ближайшую неделю
8. Редактирование информации пользователя

ГЛАВА 2. Функционал сервиса

2.1. Визуальный интерфейс

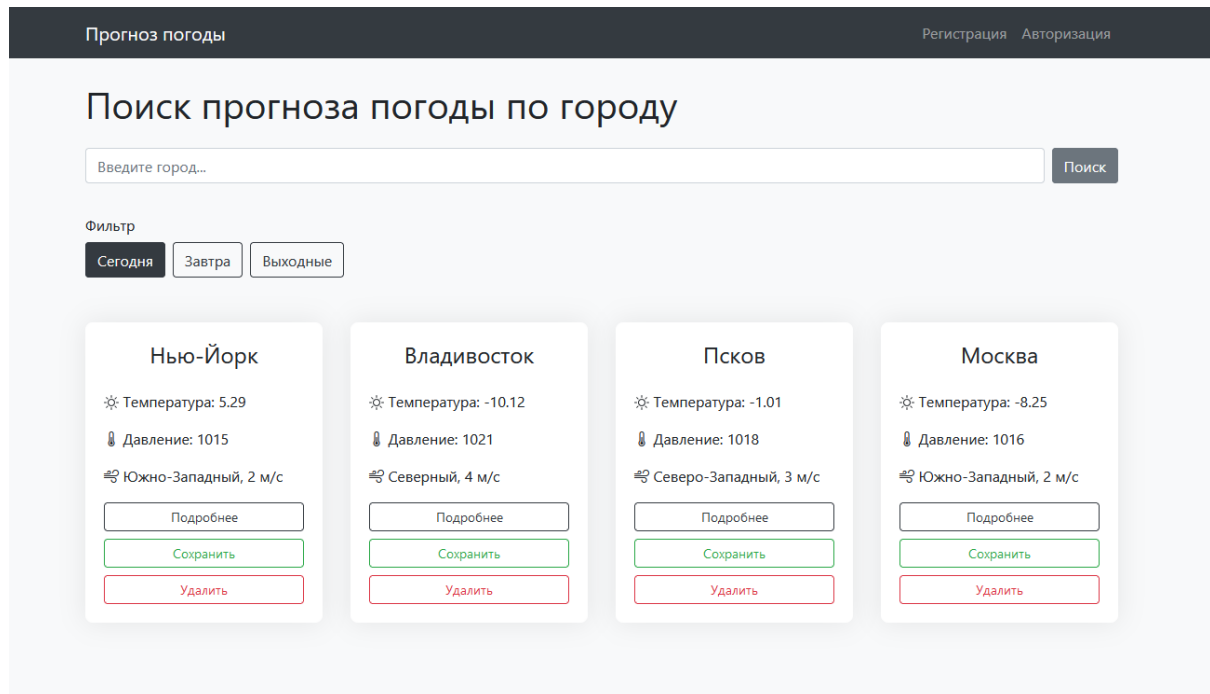


Рисунок 1. Главная страница

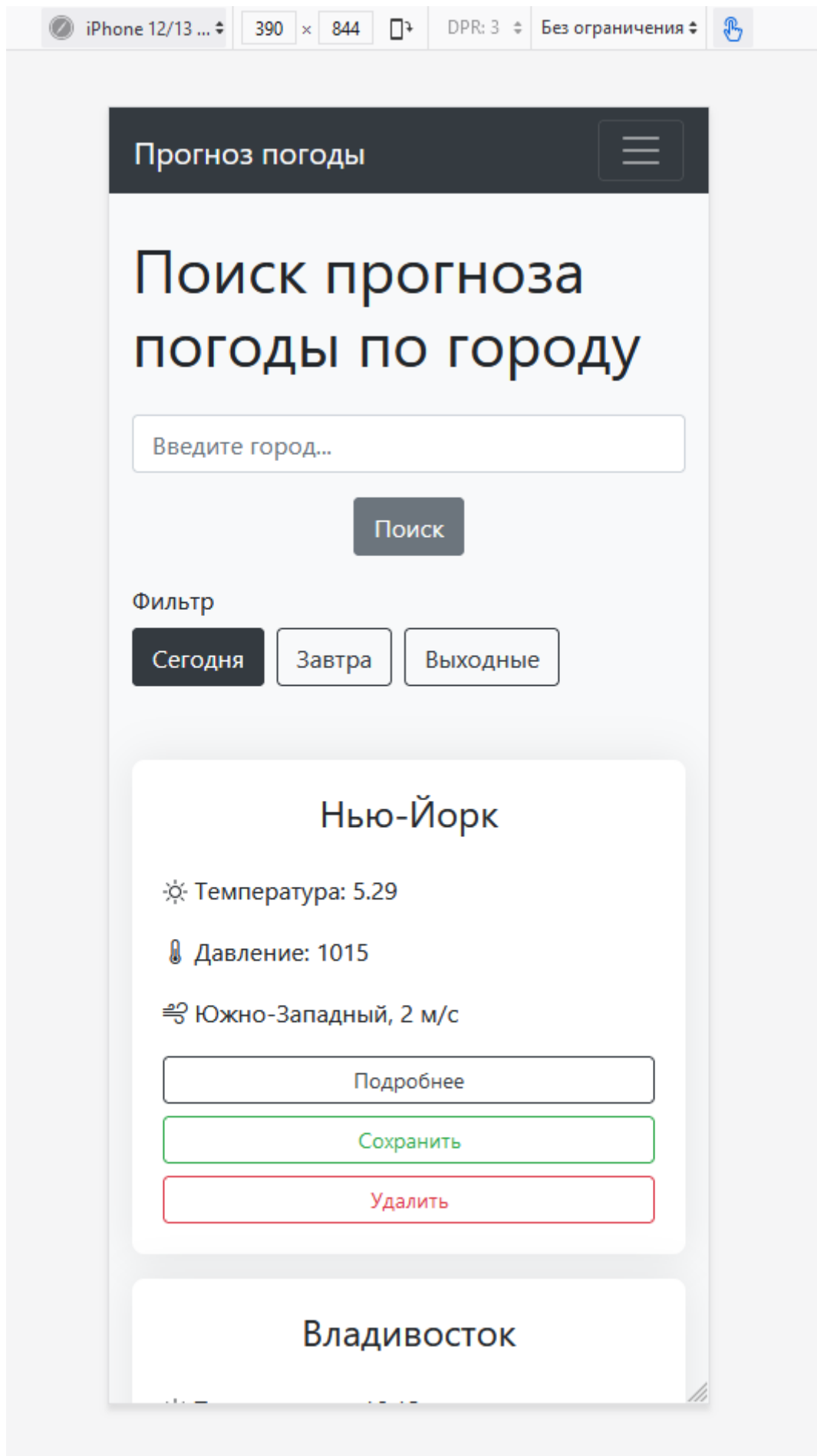


Рисунок 2. Главная страница с мобильного устройства

The screenshot shows the mobile app interface. At the top, a dark header bar contains the text "Прогноз погоды" on the left and "Регистрация Авторизация" on the right. The main content area is light gray and features a white registration form titled "Регистрация". The form has three input fields: "Email" with the placeholder "Введите email", "Логин*" with the placeholder "Введите логин", and "Пароль*" with the placeholder "Введите пароль". Below these fields is a dark button labeled "Зарегистрироваться".

Рисунок 3. Регистрация на сайте

This screenshot shows the same registration form as in Figure 3, but with error messages. The "Логин*" field, which contains the text "test", has a red error message below it: "Пользователь с таким именем уже существует." The "Пароль*" field, which contains four dots, also has a red error message below it: "Пароль должен содержать минимум 8 символов, включая прописные и строчные буквы, цифры и специальные символы." The "Зарегистрироваться" button remains at the bottom of the form.

Рисунок 4. Регистрация на сайте. Каждое поле отображает ошибки, приходящие с сервера

iPhone 12/13 ... 390 x 844 DPR: 3 Без ограничения

Прогноз погоды

Регистрация

Email

Логин*

Пароль*

Зарегистрироваться

Рисунок 5. Регистрация на сайте. Каждое поле отображает ошибки, приходящие с бекенда

Прогноз погоды

РегистрацияАвторизация

Авторизация

Логин

Введите логин

Пароль

Введите пароль

[Забыли пароль?](#)

Авторизоваться

Рисунок 6. Авторизация на сайте

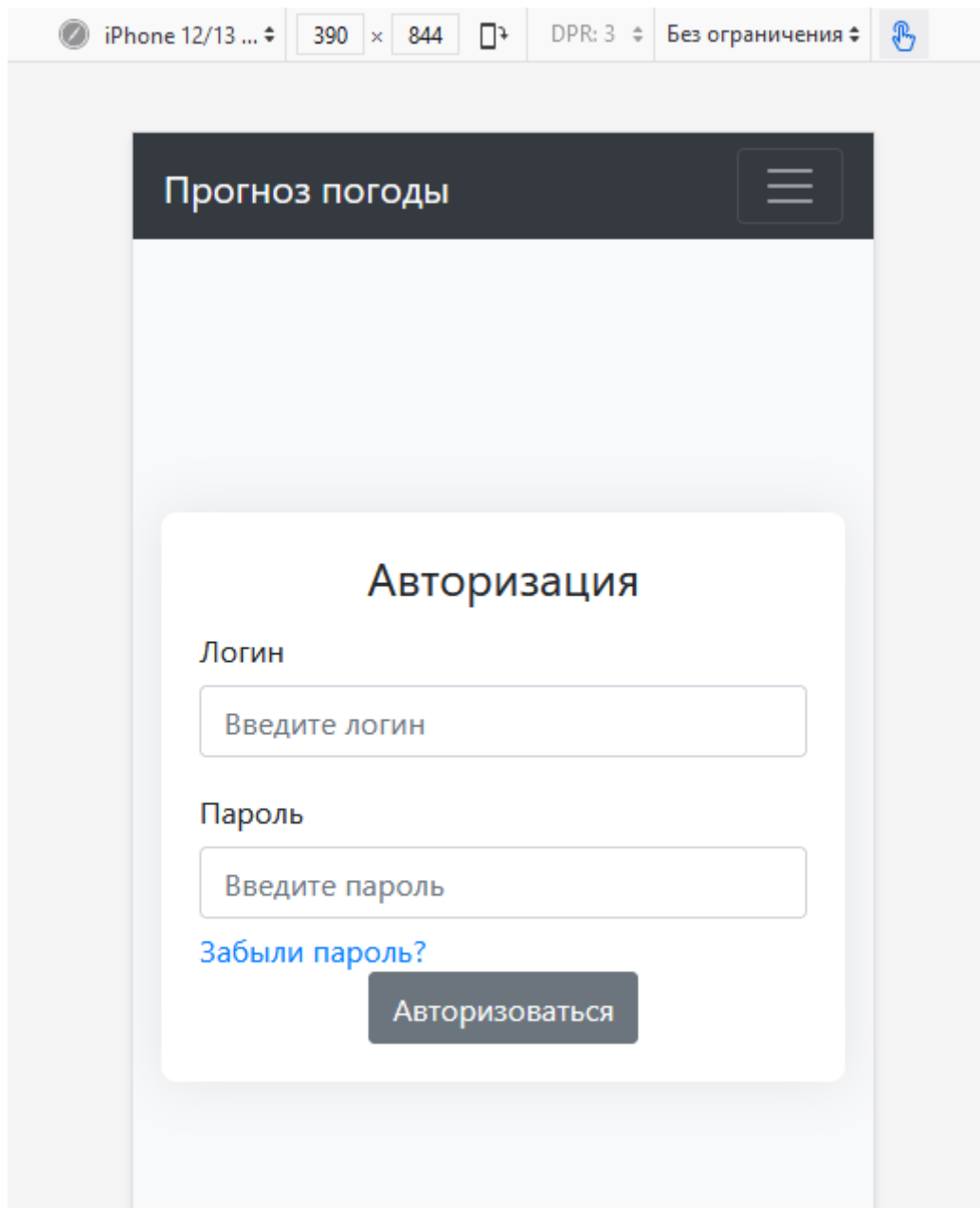


Рисунок 7. Авторизация с мобильных устройств

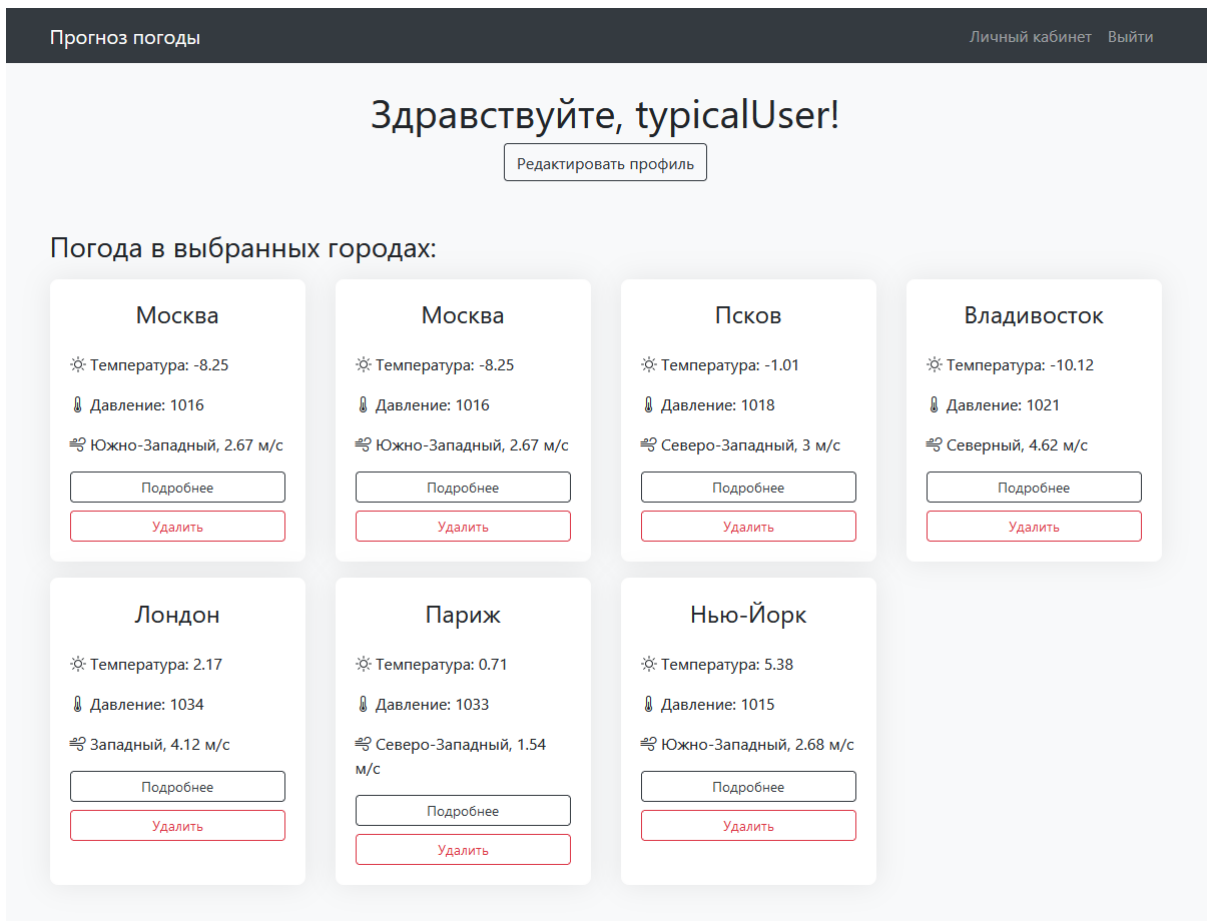


Рисунок 8. Личный кабинет

Здравствуйте,
typicalUser!

Редактировать профиль

Погода в выбранных
городах:

Москва

☀ Температура: -8.25

📏 Давление: 1016

🌀 Южно-Западный, 2.67 м/с

Подробнее

Удалить

Калуга

☀ Температура: -9.98

📏 Давление: 1015

🌀 Южно-Западный, 2 м/с

Рисунок 9. Личный кабинет с мобильных устройств

Редактирование профиля

Редактирование Email

Сохранить

Редактирование имени пользователя

Необходимо ввести текущий пароль

Сохранить

Рисунок 10. Редактирование профиля

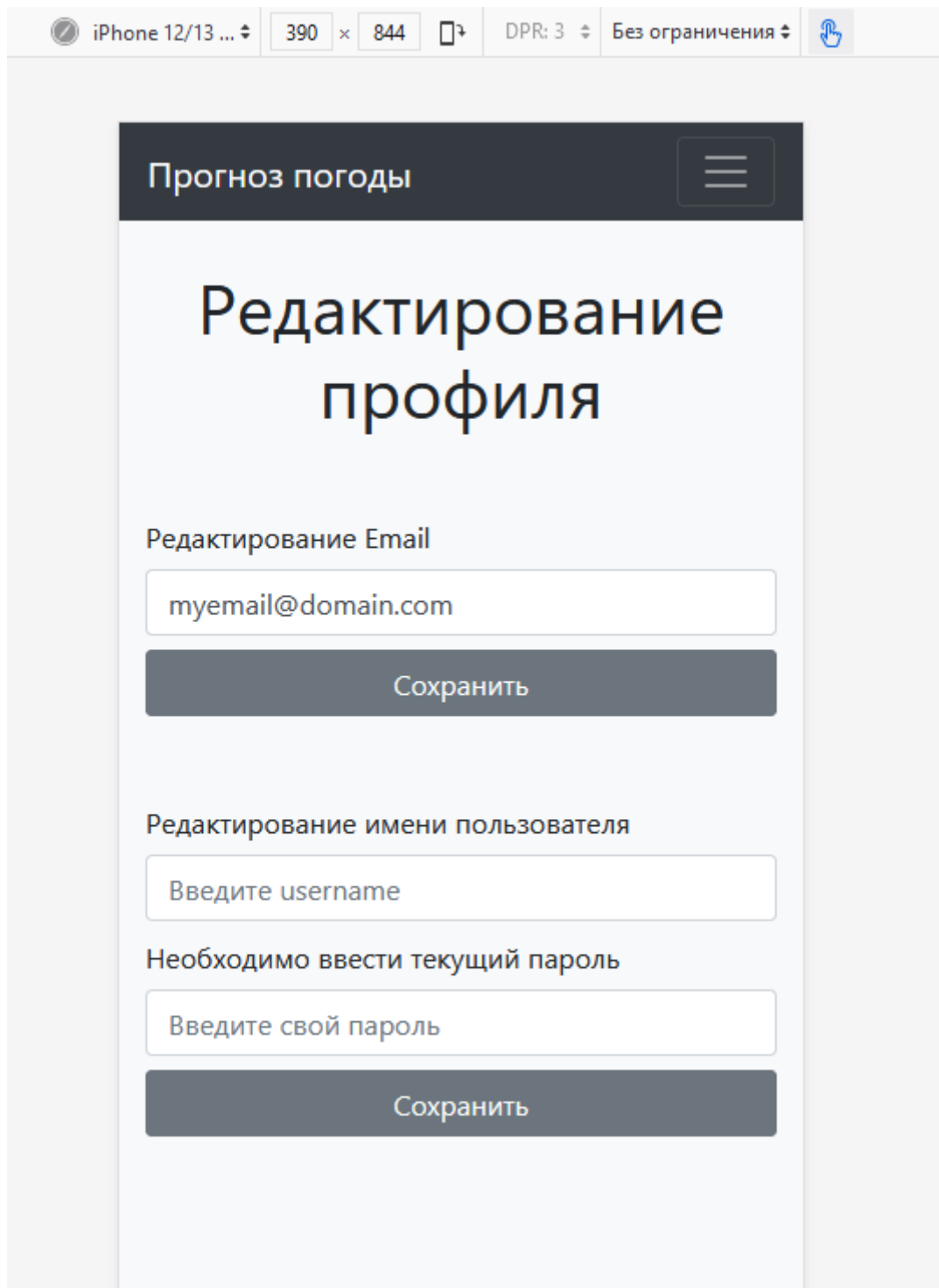


Рисунок 11. Редактирование профиля с мобильных устройств

Забыли пароль?

Введите почту, которую вы указывали при регистрации

Сбросить пароль

Рисунок 12. Сброс пароля

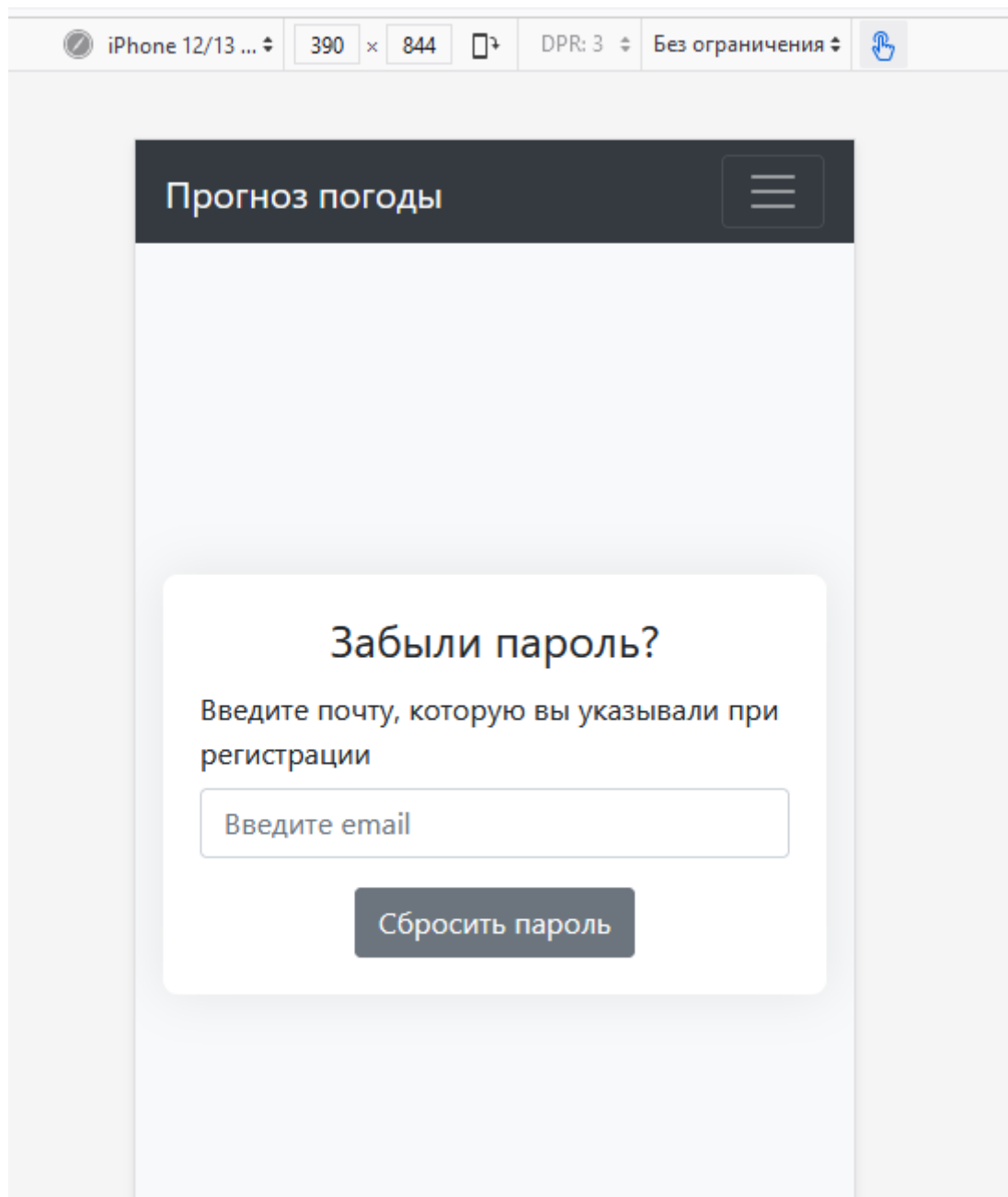


Рисунок 13. Сброс пароля с мобильных устройств

Успешно

Проверьте почту. Вы должны были получить письмо со ссылкой на сброс пароля.

Рисунок 14. Сообщение об успешном сбросе пароля

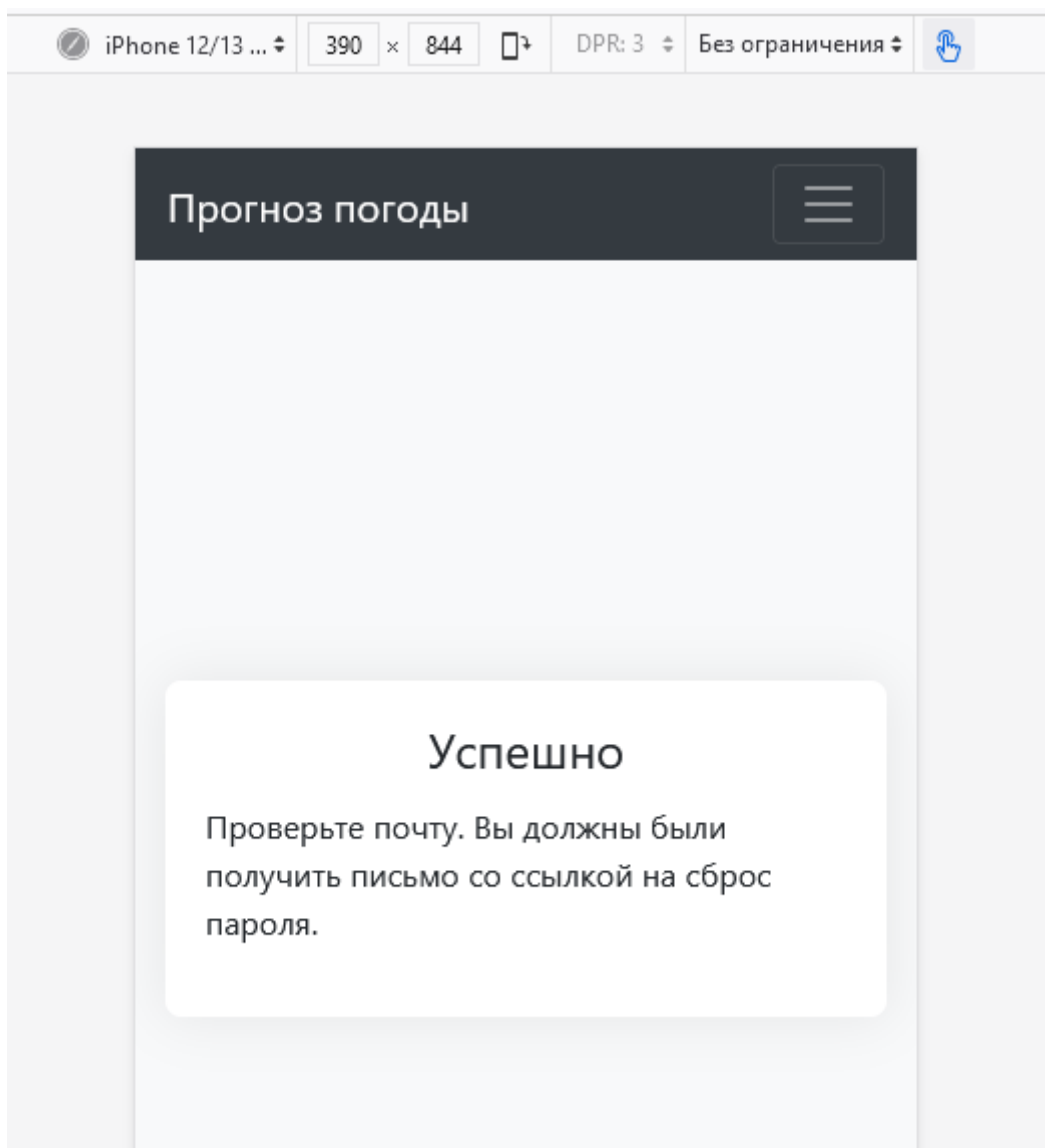


Рисунок 15. Сообщение об успешном сбросе пароля с мобильных устройств

Санкт-Петербург, -4

Погода на ближайшую неделю:

25.01 ☀ Температура: -2.29 📏 Давление: 1014 🌀 Южно-Западный, 6.38 м/с	26.01 ☀ Температура: -6.74 📏 Давление: 1016 🌀 Южный, 4.15 м/с	27.01 ☀ Температура: -3.66 📏 Давление: 998 🌀 Юго-Восточный, 4.81 м/с	28.01 ☀ Температура: -3.01 📏 Давление: 997 🌀 Северо-Восточный, 4.37 м/с
29.01 ☀ Температура: -10.01 📏 Давление: 1009 🌀 Юго-Восточный, 4.4 м/с	30.01 ☀ Температура: -2.29 📏 Давление: 975 🌀 Юго-Восточный, 8.7 м/с	31.01 ☀ Температура: -9.67 📏 Давление: 998 🌀 Северо-Восточный, 4.77 м/с	1.02 ☀ Температура: -13.55 📏 Давление: 1007 🌀 Северо-Западный, 2.3 м/с

Рисунок 16. Прогноз на неделю

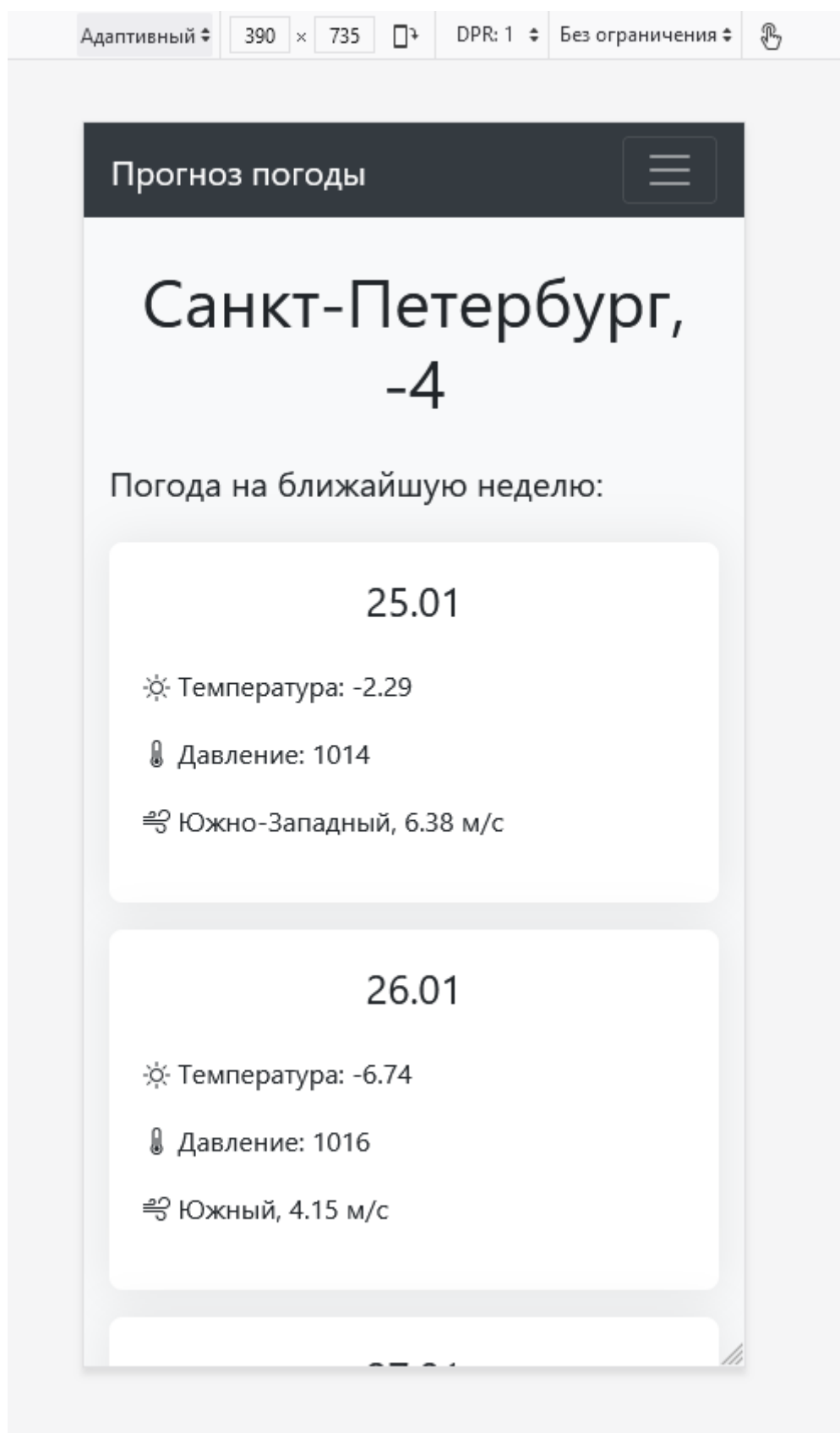


Рисунок 17. Прогноз на неделю с мобильных устройств

2.2. Код

```
async resolveCityName(city) {
  let result = { status: true, data: null, error: null }

  let url = `http://127.0.0.1:8000/city/?search=${city}`

  try {
    const response = await this.axios.get(url)

    if (response.data.length === 0) {
      result.status = false
      result.error = 'Ошибка! Указанный город не найден'
    } else {
      result.data = response.data
    }
  } catch (e) {
    result.status = false
    result.error = 'Ошибка! Указанный город не найден'
  }

  return result
},
```

Рисунок 18. Получение информации о городе с сервера с обработкой возможных ошибок

```

async apiCityWeather(city, date) {
  let result = { status: true, data: null, error: null }

  let cityResolved = await this.resolveCityName(city)

  let url = `http://127.0.0.1:8000/weather/?city=${cityResolved.data.id}`

  try {
    const response = await this.axios.get(url)

    if (date === 'today') {
      result.data = response.data.current
    } else if (date === 'tomorrow') {
      result.data = response.data.daily[1]
      result.data.temp = result.data.temp.day
    } else if (date === 'weekends') {
      let today = new Date().getDay()
      let weekend = 7
      let diff = weekend - today

      if (diff === 0) {
        diff = 7
      }

      result.data.weatherid = result.data.id
      result.data = response.data.daily[diff]
      result.data.temp = result.data.temp.day
    }

    result.data.weatherid = response.data.id
    result.data.cityid = cityResolved.data.id
    result.data.name = cityResolved.data.local_names.ru
  } catch (e) {
    console.log('error getting weather', e)
    result.status = false
    result.error = 'Ошибка! Не удалось получить прогноз погоды'
  }
}

```

Рисунок 18. Получение информации о погоде и фильтрация по указанной дате - текущая погода, на завтра или на ближайшие выходные

```

1 <template>
2 <div class="home-page">
3   <b-container>
4     <h1 class="mt-4">Поиск прогноза погоды по городу</h1>
5
6     <b-form @submit.prevent="getWeather">
7       <div class="d-sm-flex align-items-start mt-4">
8         <b-form-group class="flex-grow-1">
9           <b-form-input v-model="filterCity"
10             placeholder="Введите город..."
11             :state="filterError === null ? null : false"
12           />
13           <b-form-invalid-feedback>
14             {{ filterError }}
15           </b-form-invalid-feedback>
16         </b-form-group>
17
18         <div class="d-flex justify-content-center mt-2 mt-sm-0 ml-sm-2">
19           <b-button type="submit">Поиск</b-button>
20         </div>
21       </div>
22       <b-form-group label="Фильтр" class="mt-3">...</b-form-group>
23     </b-form>
24
25     <b-row v-if="weatherInCities.length !== 0" cols="1" cols-md="2" cols-lg="3" cols-xl="4" class="mt-5">
26       <b-col v-for="city in weatherInCities" class="mb-3">
27         <weather-city :id="city.cityid"
28           :weatherid="city.weatherid"
29           :title="city.city"
30           :weather="city.weather"
31           :pressure="city.pressure"
32           :wind="city.wind"
33           :wind-direction="city.windDirection"
34           :save-button="true" :delete-button="true"
35           @save-city="saveCity"
36           @delete-city="deleteCity"
37         />
38       </b-col>
39     </b-row>
40
41     <h5 v-else class="mt-4">
42       Введите название города чтобы добавить карточку с погодой
43     </h5>

```

Рисунок 18. JSX

Выводы по работе

Мы создали персонализированный сервис по прогнозу погоды с возможностью добавлять в личный кабинет города, за которым я хочу следить. Мои навыки в Vue.js значительно улучшились. Я закрепила свои знания по работе с открытым API и запросами на бекенд. Цель курсовой работы выполнена

СПИСОК ЛИТЕРАТУРЫ

1. Документация Vue.js [Электронный ресурс] — <https://ru.vuejs.org/v2/guide/>
Дата обращения 23.01.2022.
2. Открытое API OpenWeather для получение данных прогноза погоды в разных городах [Электронный ресурс] — <https://openweathermap.org/api> Дата обращения 23.01.2022.
3. Документация Vue.js Router [Электронный ресурс]
— <https://router.vuejs.org/ru/guide/> Дата обращения 23.01.2022.