



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

---

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Компьютерные системы и сети

**О Т Ч Е Т**

**по лабораторной работе № 7**

**Название:** Основы Front-End разработки на JavaScript

**Дисциплина:** Языки интернет-программирования

Студент	<u>ИУ6-32Б</u> (Группа)	<u>12.11.2024</u> (Подпись, дата)	<u>Т.А. Гаджиев</u> (И.О. Фамилия)
Преподаватель		<u>12.11.2024</u> (Подпись, дата)	<u>В.Д. Шульман</u> (И.О. Фамилия)

Москва, 2024

Цель работы — изучение основ разработки SPA-приложение на JavaScript.

Ход работы.

1. Ознакомились с материалами для подготовки перед выполнением лабораторной работы.
2. Сделали форк данного репозитория в GitHub, клонировали получившуюся копию локально, создали от мастера ветку дев и переключились на нее:

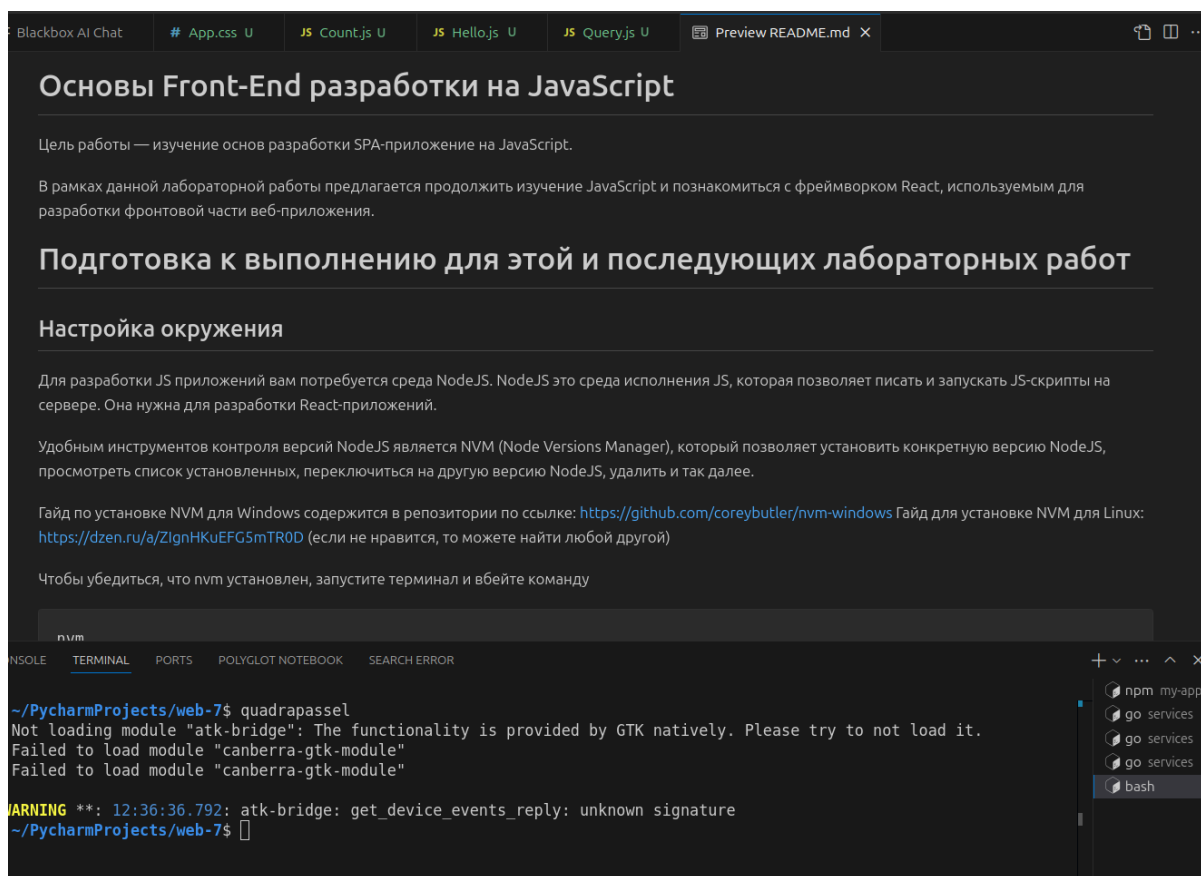


Рисунок 1 — Скопированный репозиторий

3. Написали 3 компонента для React проекта. Код серверов и результаты запросов в Postman прикрепили ниже:

Компонент(Вывод строки):

```
import React, { useState } from 'react';

const Hello = () => {
  const [message, setMessage] = useState("");
  const fetchHelloMessage = async () => {
```

```

try {
  const response = await fetch('http://localhost:8081/getMessage');
  if (!response.ok) {
    throw new Error('Network response was not ok');
  }
  const data = await response.text();
  setMessage(data);
} catch (error) {
  console.error('Fetch error:', error);
  setMessage('Error fetching message');
}
};

return (
  <div>
    <h2>Get Hello Message</h2>
    <button onClick={fetchHelloMessage}>Получить привет</button>
    <p>Сообщение: {message}</p>
  </div>
);
};

export default Hello;

```

## Компонент 2(Вывод строки с ключом):

```

import React, { useState } from 'react';

const User = () => {
  const [name, setName] = useState('');
  const [greeting, setGreeting] = useState('');

  const fetchUserGreeting = async () => {
    try {
      const response = await fetch(`http://localhost:8082/api/user?name=${name}`);
      if (!response.ok) {
        throw new Error('Network response was not ok');
      }
      const data = await response.text();
      setGreeting(data);
    } catch (error) {
      console.error('Fetch error:', error);
      setGreeting('Пожалуйста, введите Ваше имя');
    }
  };

  return (
    <div>
      <h2>Знакомство</h2>
      <input

```

```

type="text"
placeholder="Введите своё имя"
value={name}
onChange={(e) => setName(e.target.value)}
/>
<button onClick={fetchUserGreeting}>Поприветствовать</button>
<p>Сообщение: {greeting}</p>
</div>
);
};

export default User;

```

### Компонент 3(Count):

```

import React, { useState } from 'react';

const Count = () => {
  const [count, setCount] = useState(0);

  const fetchCount = async () => {
    try {
      const response = await fetch('http://localhost:8083/count');
      if (!response.ok) {
        throw new Error('Network response was not ok');
      }
      const data = await response.text();
      setCount(data);
    } catch (error) {
      console.error('Fetch error:', error);
      setCount('Error fetching count');
    }
  };

  const incrementCount = async () => {
    try {
      await fetch('http://localhost:8083/count', {
        method: 'POST',
        headers: {
          'Content-Type': 'application/x-www-form-urlencoded',
        },
        body: new URLSearchParams({ count: 1 }),
      });
    } catch (error) {
      console.error('Fetch error:', error);
    }
  };

  return (
    <div>

```

```

<h2>Count Service</h2>
<button onClick={fetchCount}>Счётчик</button>
<button onClick={incrementCount}>Увеличить счётчик</button>
<p>Счётчик: {count}</p>
</div>
);
};

export default Count;

```

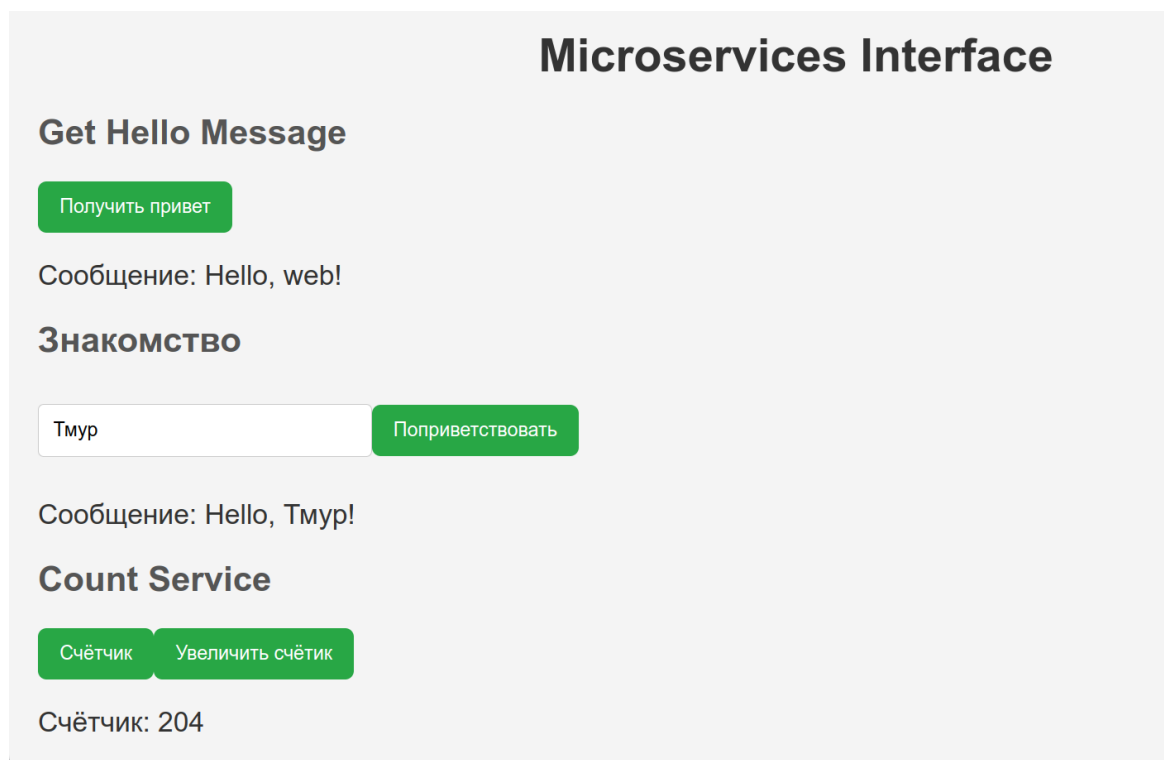


Рисунок 2 — Интерфейс React приложения

4. Зафиксировали изменения, сделали коммит и отправили полученное состояние ветки dev в удаленный репозиторий GitHub. Через интерфейс GitHub создали Pull Request dev --> master

Заключение: в ходе лабораторной работы изучили JavaScript и познакомились с фреймворком React, используемым для разработки фронтонвой части веб-приложения.