**3.1 Типы данных –** изучено

**Упражнение 1**

**Постановка задачи:**

Создайте react-приложение с помощью Create React App используя npx и запустите сборку.

**Выполнение:**

Было создано react приложение с использованием Vite, так как CRA выдавал ошибку (рис. 4.1). Запущенное приложение показано на (рис 4.2)

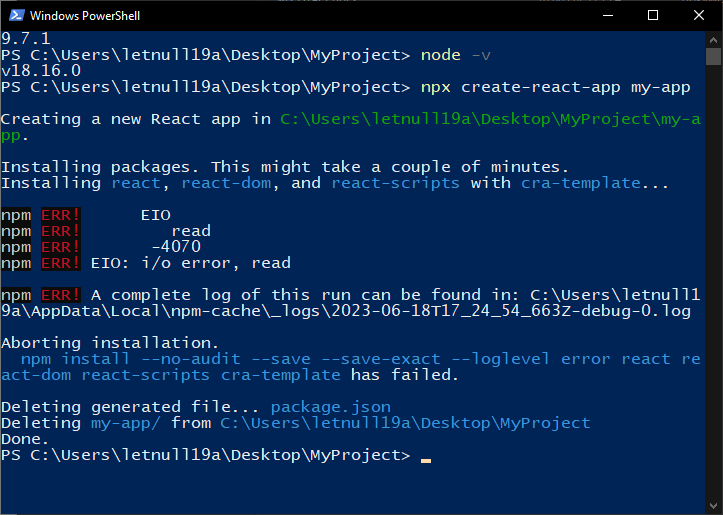


Рисунок 4.1 – Запущенное приложение

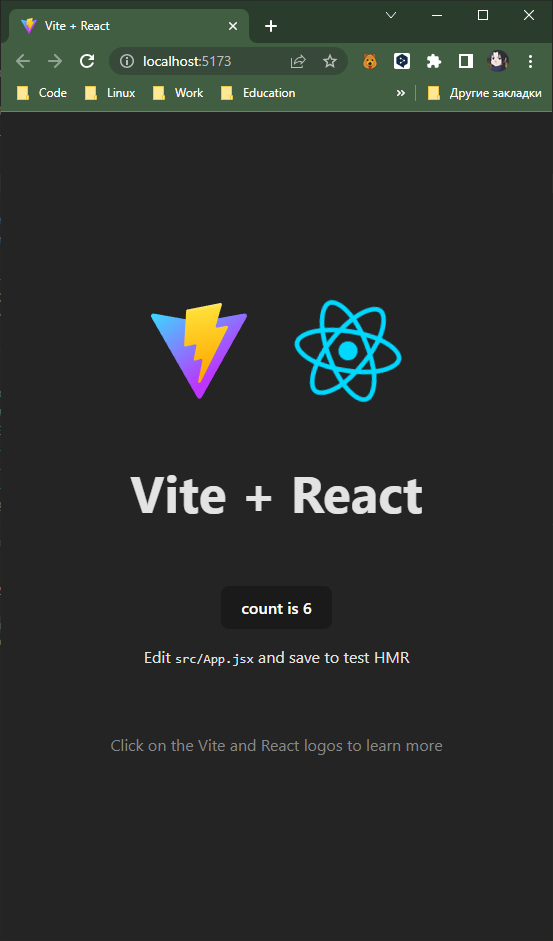


Рисунок 4.2 – Ошибка

**Упражнение 2**

**Постановка задачи:**

Создайте компонент, который будет выводить ваше имя в заголовке. Импортируйте новый компонент в App.js файл.

**Выполнение:**

Был создан компонент React который отображал вводимое в него имя (рис. 4.3). Код компонента представлен в листинге 4.1.

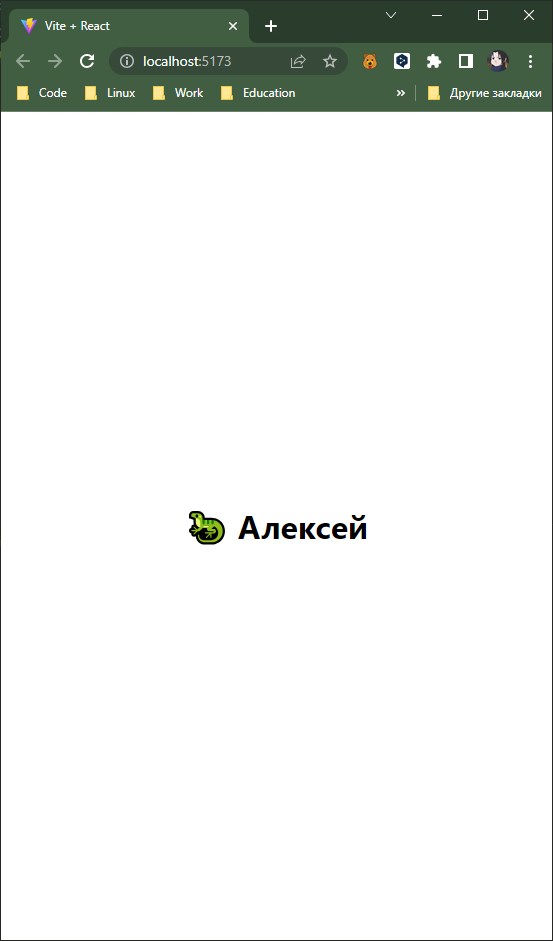


Рисунок 4.3 – Выполнение кода

Листинг 4.1 ­ Код компонента

export const Heading = () => <h1>&#129422; Алексей</h1>

**Упражнение 3**

**Постановка задачи:**

Создайте новый компонент для отображения имя пользователя, возраст и город. '

**Выполнение:**

Был добавлен компонент который выводит данные из props (рис. 4.5).

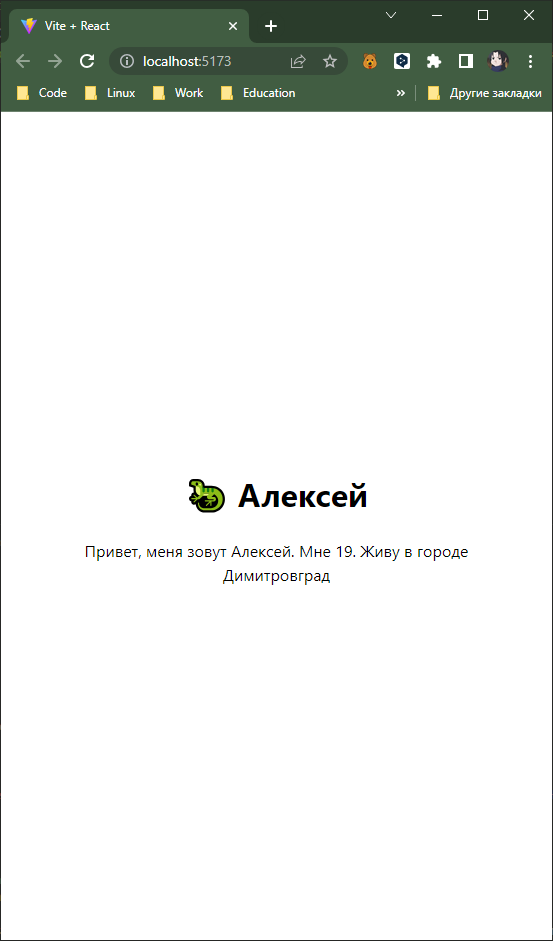


Рисунок 4.5 – Выполнение кода на странице

**Упражнение 4**

**Постановка задачи:**

Создайте компонент, который будет выводить текст в заголовке при вводе символов в элемент Input.

**Выполнение:**

Был создан компонент представляющий из себя поле ввода, при вводе символов который отобразил бы значение в заголовке. Задача была решена двумя способами. Первый заключался в использовании состояния, второй через хук ссылки. Во втором способе не обошлось без изменения состояния. Результат представлен в (рис. 4.6).

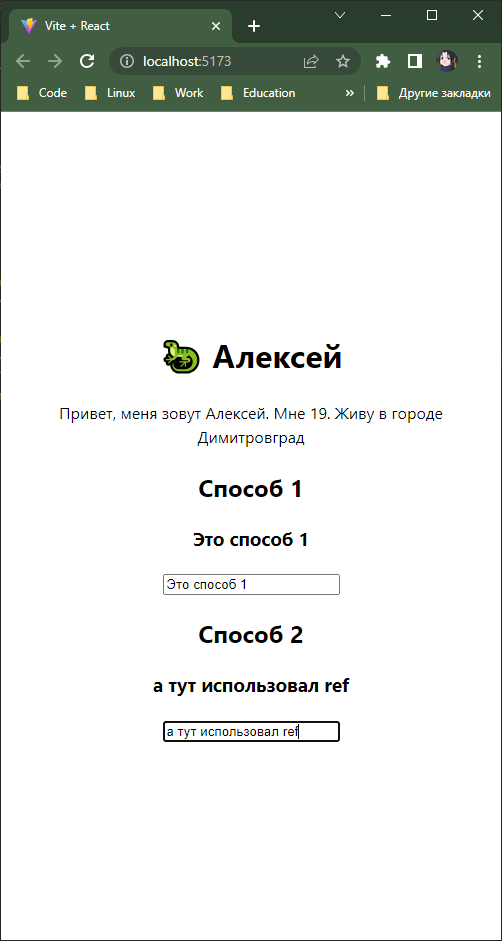


Рисунок 4.6 – Вывод данных

**Упражнение 5**

**Постановка задачи:**

Создайте компонент, который в консоли браузера будет выводить общее количество рендеров. Добавьте два состояния text и count. Состояние count изменяется при нажатие на кнопку, а text при вводе символов в поле ввода.

**Выполнение:**

Был дополнен компонент App, в него были внесены новые состояния, в том числе и состояние перерисовки которое фиксировало сколько внесено изменений в состояниях (рис. 4.7).

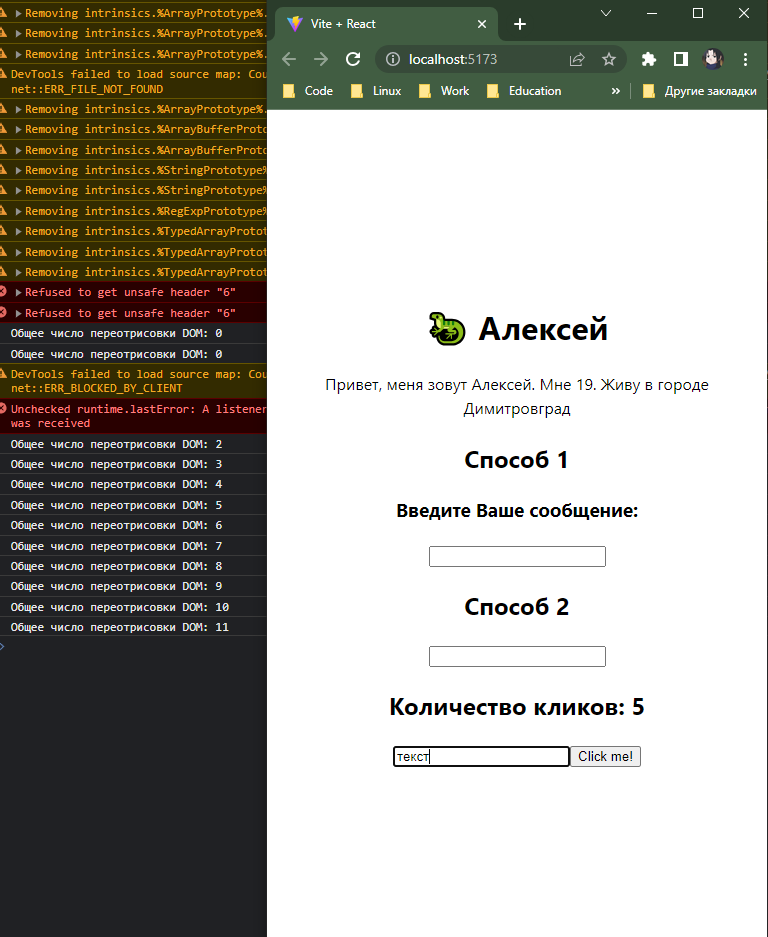


Рисунок 4.7 – Выполнение кода

**Упражнение 6**

**Постановка задачи:**

Создайте цепочку из 5 компонентов, в родительском компоненте будет отображаться количество кликов по кнопке. В последнем компоненте создайте кнопку и через контекст передайте ей функцию для изменения состояния количество кликов.

**Выполнение:**

Были созданы дополнительно пять компонентов, каждый из которых вложен в соседний. Был объявлен контекст. В него был помещён метод для изменения состояния и само состояние. Контекст менялся в последнем компоненте при нажатии на кнопку. (рис. 4.8).

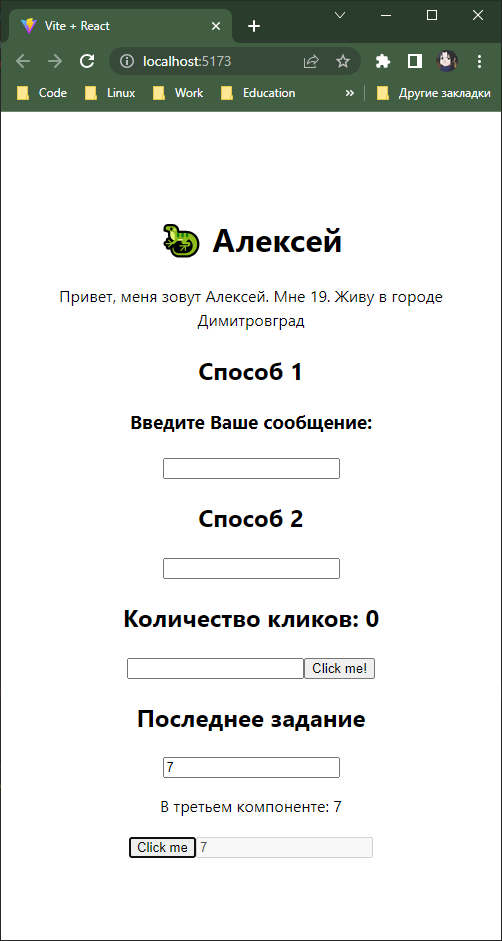


Рисунок 4.8 – Выполнение кода