

УДК 004.77

## ПРЕИМУЩЕСТВА ИСПОЛЬЗОВАНИЯ БИБЛИОТЕКИ REACT ПРИ РАЗРАБОТКЕ WEB-ПРИЛОЖЕНИЯ

И. А. БЕККЕР, К. С. ЛАЛОВ  
Белорусско-Российский университет  
Могилев, Беларусь

Приложение «Zdorovo!» разрабатывается в сфере экологии человека студентами 4 курса группы АСОИР-181 с привлечением талантливых студентов 3 курса. Изначально было задумано вести в приложении контроль потребляемых калорий и микронутриентов с учетом цели, заявленной пользователем, но функционал со временем разросся, несколько изменился, изменился и набор средств программирования.

На данный момент приложение «Zdorovo!» представляет собой web-сайт по поддержанию здорового образа жизни с серверной и клиентской частью. Серверная часть реализует на языке Python следующий функционал: организацию доступа пользователя к личной учетной записи, хранение личных данных пользователя.

Для разработки клиентской (внешней) части web-приложения было решено в итоге использовать JavaScript-библиотеку React.js. Основные причины, по которым разработчики остановились на этом инструменте программирования: проект «Zdorovo!» содержит большое число динамических страниц и повторяющегося кода, что позволило использовать все плюсы библиотеки React.js. Именно фрагменты повторяющегося кода, соответствующие одному из элементов приложения, являются особенностью этой библиотеки. Они называются компоненты, а их различный внешний вид описывается в React.js таким понятием, как состояния компонентов. Состояние State – это информация об изменяемом отображении компонента React. Для простого тиражирования компонента React используется другой тип данных Props.

За счет такой структуры программы получилось меньше кода, написанного «руками». Сократило время и размер кода и то, что использовались вложенные компоненты.

Проект «Zdorovo!» получился сравнительно небольшой и не потребовал специальных инструментов для работы с состояниями, что не дало проявиться такому минусу библиотеки React, как сложность взаимодействия между компонентами разных уровней вложения.

Выбор компонентов был осуществлен исходя из степени повторяемости фрагмента кода, соответствующего описанию объекта, в качестве компонентов взяты карточка продукта питания, параметры настроек исходных данных пользователя, компоненты выбора цели.

Для написания HTML-верстки в .js-файлах React использует препроцессор JSX со специальным синтаксисом, что позволяет библиотеке создавать компоненты, содержащие внутри себя и HTML-разметку, и логику.

На языке JSX сначала было создано окно входа с предусмотренными компонентами: название приложения и другие надписи, поля ввода логина и пароля, кнопка отправки данных на сервер (рис. 1).

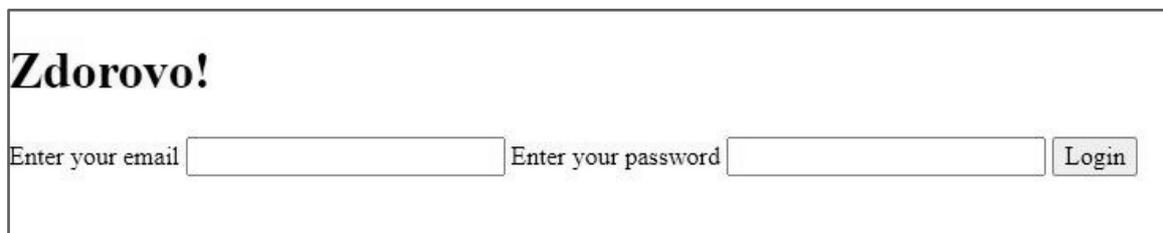
```

1  import React from "react";
2
3  export default function Registration() {
4    return (
5      <section className="login">
6        <h1 className="login_title">Food For Pleasure</h1>
7        <form className="login_form" action="#">
8          <label className="form_label" htmlFor="user-login">
9            Enter your login
10         </label>
11         <input className="form_input" id="user-login" type="text" />
12         <label className="form_label" htmlFor="user-password">
13           Enter your password
14         </label>
15         <input
16           className="form_input form_input_last"
17           id="user-password"
18           type="password"
19         />
20         <input className="form_submit" value="Login" type="submit" />
21       </form>
22     </section>
23   );
24 }

```

Рис. 1. Функция, описывающая компонент Registration на JSX

В результате первого этапа работы с компонентами в окне входа в приложение «Zdorovo!» появились все необходимые элементы. На этом этапе разработки у них еще отсутствовала стилизация (рис. 2).



**Zdorovo!**

Enter your email  Enter your password

Рис. 2. Результат создания окна входа

Стилизация элементов (размеры, формат фона и границ) является следующим этапом написания программного кода (рис. 3).

```
1  .login {  
2    display: flex;  
3    flex-direction: column;  
4  
5    padding: 180px 30px 120px 30px  
6    box-sizing: border-box;  
7  }  
8  
9  .login__title {  
10   margin: 0 0 40px 0;  
11  
12   font-size: 2.2em;  
13   text-align: center;  
14 }
```

Рис. 3. Фрагмент описания стилей

По завершении этого этапа стилизованные элементы стали отображаться в браузере уже не как компоненты библиотеки React.js, а именно как html-элементы (рис. 4).



Рис. 4. Результат применения стилей к HTML-элементам

Использование React.js при программировании web-приложения «Zdorovo!» позволило в гибком, структурированном виде представить код приложения, легко описывать, изменять состояния компонентов, а также выявлять и исправлять ошибки по тем компонентам, которые некорректно работали. С помощью компонентного подхода достигнута заметно более высокая производительность за счет не полного, как в JavaScript и HTML, а частичного перестраивания дерева элементов.