**ОТЧЕТ**

по лабораторной работе №7

по курсу «Разработка Web-приложений»

Выполнили:

студенты группы 16ВВ2

Аккуратнов И.М.

Кожаев В.Д.

Принял:

Дубравин А.В.

Пенза 2020

Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

### Название

### 7. Frontend на ReactJS

### Цель работы

Необходимо для лабораторной работы №5 реализовать клиентское приложение с использованием технологи ReactJS.

Необходимо в интерфейсе предусмотреть элементы управления для всех параметров, которые можно передать на сервер через REST API. Нужно не забыть, что часть параметров может быть опциональной.

Так же нужно реализовать кнопку, которая должна активировать запрос к серверу и по результатам ответа изменить состояние элементов управления в соответствии с полученными значениями, а также заполнить поле с именем изображения и вместе с этим необходимо выполнить второй запрос к серверу, загрузить изображение по его имени и отобразить на странице.

Вариант Задание

1. Животные: антилопа, акула, ара.

### Листинг

**Файл** *App.js*

*import* React, {*useReducer*, *useState*} from "react";  
*import* 'babel-polyfill';  
  
*const* initialState = {  
 name: '',  
 width: '',  
 height: '',  
 colour: ''  
};  
  
*const* CHANGE\_PICTURE\_NAME = 'main/CHANGE\_PICTURE\_NAME';  
*const* CHANGE\_PICTURE\_WIDTH = 'main/CHANGE\_PICTURE\_WIDTH';  
*const* CHANGE\_PICTURE\_HEIGHT = 'main/CHANGE\_PICTURE\_HEIGHT';  
*const* CHANGE\_PICTURE\_COLOUR = 'main/CHANGE\_PICTURE\_COLOUR';  
  
*const* reducer = (state, action) => {  
 *console*.log({state, action});  
 *switch*(action.type) {  
 *case* CHANGE\_PICTURE\_NAME:  
 *return* { ...state, name: action.payload };  
 *case* CHANGE\_PICTURE\_WIDTH:  
 *return* { ...state, width: action.payload };  
 *case* CHANGE\_PICTURE\_HEIGHT:  
 *return* { ...state, height: action.payload };  
 *case* CHANGE\_PICTURE\_COLOUR:  
 *return* { ...state, colour: action.payload };  
 *default*:  
 *return* state;  
 }  
};  
  
*const* request = *async* (state) => {  
 *let* response = *await fetch*('http://localhost:8888/upload', {  
 method: 'POST',  
 mode: 'no-cors',  
 headers: {  
 'Content-Type': 'application/x-www-form-urlencoded',  
 },  
 body: `name=${state.name}&width=${state.width}&height=${state.height}&colour=${state.colour}`  
 });  
  
 *return await* response.blob();  
};  
  
*const component* = () => {  
 *const* [state, dispatch] = *useReducer*(reducer, initialState);  
 *const* [image, setImage] = *useState*(*null*);  
  
 *const* getImage = *async* (event) => {  
 event.preventDefault() && event.stopPropagation();  
 *const* picture = *await* request(state);  
 *console*.log(picture);  
 setImage(picture);  
 };  
  
 *return*(  
 <div>  
 <div>  
 <img *src*={image} *alt*="picture" />  
 </div>  
 <form>  
 Введите название картинки с номером  
 <input *onChange*={ event => dispatch({type: CHANGE\_PICTURE\_NAME, payload: event.target.value})} *name*="name" *type*="text" *required pattern*="(shark|antelope|macaw)( [1-9])?$"  
 *title*="Можно ввести только shark, antelope, macaw с цифрой от 1 до 9"/>  
 <br/>  
 Введите ширину  
 <input *onChange*={ event => dispatch({type: CHANGE\_PICTURE\_WIDTH, payload: event.target.value})} *name*="width" *type*="text" *min* = "100" *max* = "1500"  
 *title*="Диапазон от 100 до 1500 пикселей"/>  
 Введите высоту  
 <input *onChange*={ event => dispatch({type: CHANGE\_PICTURE\_HEIGHT, payload: event.target.value})} *name*="height" *type*="text" *min* = "100" *max* = "1500"  
 *title*="Диапазон от 100 до 1500 пикселей"/>  
 <br/>  
 Значения цвета  
 <input *onChange*={ event => dispatch({type: CHANGE\_PICTURE\_COLOUR, payload: event.target.value})} *name*="colour" *type*="text" *pattern*="(blue|red|green)?$"  
 *title*="Можно ввести только blue, red, green"/>  
 <br/>  
 <input *onClick*={(event) => getImage(event)} *type*="submit" *value*="Submit" />  
 </form>  
 </div>  
 )  
};  
  
*export default component*;

**Файл** *index..html*

<!DOCTYPE *html*>  
<html *lang*="en">  
<head>  
 <meta *charset*="UTF-8">  
 <title>Lab7</title>  
</head>  
<body>  
 <div *id*="app"></div>  
</body>  
</html>

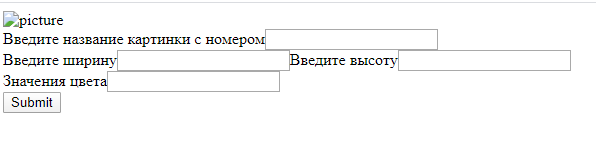
**Файл** *index.js*

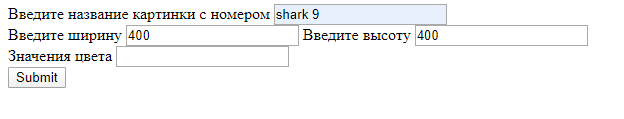
*import* React from 'react';  
*import* ReactDOM *from* 'react-dom';  
  
*import APP from* './App'  
  
*const* App = () => {  
 *return*(  
 <div>  
 <APP />  
 </div>  
 )  
};  
  
ReactDOM.render(<App />, document.querySelector('#app'));

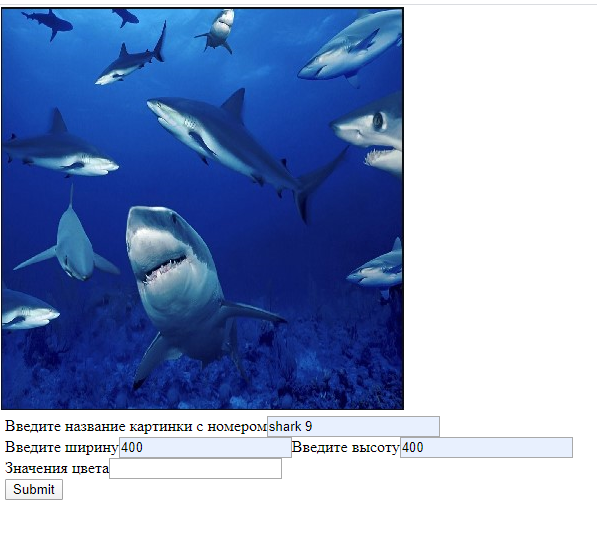
**Файл** *request.js*

*function request*(url, options, skipJsonParse) {  
 *let* headers;  
 *return fetch*(url, options)  
 .then((response) => {  
 *if* (skipJsonParse) {  
 *return* response.text().then(  
 data => {  
 *if* (response.ok) {  
 *return* data;  
 } *else* {  
 *return* Promise.reject({ status: response.status, message: response.statusText, data });  
 }  
 }  
 );  
 } *else* {  
 *return* response.text()  
 .then(text => text ? JSON.parse(text) : {})  
 .then(data => {  
 *if* (response.ok) {  
 *return* data;  
 } *else* {  
 *return* Promise.reject({ status: response.status, message: response.statusText, data });  
 }  
 });  
 }  
 })  
 .then((data) => ({ data, headers }))  
 .catch((err) => ({  
 err  
 }));  
}  
  
*export default request*;

1. Окно поиска



1. Первый вариант запроса. Ввести название животного и номер картинки в первую строку. Ввести ширину и высоту, если хотите не стандартный размер. 



### Выводы

### В ходе выполнения лабораторной работы реализовали Frontend на React js.