Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

Лабораторная работа № 7

**«Frontend на ReactJS »**

## Отчёт по дисциплине

**«Web-программирование»**

Пенза 2020

**Выполнили с-ты гр. 16ВВ1:**

Ежов И.С.  
Екименко Д.В.

**Приняли:**

к.т.н. Дубравин А.В.  
Кармышева Н.С.

**Цель работы.**

Реализовать Frontend на ReactJS.

**Задание.**

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

|  |  |
| --- | --- |
| 5. | Животные: дельфин, дикобраз, дрозд. |

**Листинг.**

**index.js**

const express = require("express");

const mongoose = require("mongoose");

const bodyParser = require("body-parser");

const app = express();

mongoose.connect("mongodb://localhost/animal-db",

{

    useNewUrlParser: true,

    useUnifiedTopology: true

});

app.use('/images',express.static('images'));

app.use(bodyParser.json());

app.use("/api", require("./api"));

app.listen(8888, () =>{

    console.log("server start")

});

**api.js**

var fs = require("fs");

const express = require("express");

const router = express.Router();

const Animal = require("./animal")

let jsonbody = null;

let imgerror = null;

router.get("/animal", (req, res)=>{

    console.log(jsonbody);

    res.send(jsonbody);

});

router.post("/findanimal", (req, res)=>{

    console.log(req.body);

    var name = req.body.name;

    var value = req.body.value;

    if (value === '')

    {

        value = Math.floor(Math.random() \* 9);

    }

    Animal.findOne({name: name, value: value})

        .then(animal => {

            if (animal != null)

            {

                res.send({name: animal.name, value: animal.value, width: animal.width, height: animal.height, color: animal.color});

            }

            else

            {

                imgerror = {error: "Image not found"};

                res.send(imgerror);

            }

        })

});

router.post("/animal", (req, res)=>{

    console.log(req.body);

    var name = req.body.name;

    var value = req.body.value;

    var height = req.body.height;

    var width = req.body.width;

    var color = req.body.color;

    if (width == "")

    {

        width = Math.floor(Math.random() \* 2000);

    }

    if (height == "")

    {

        height = Math.floor(Math.random() \* 2000);

    }

    if (color == "")

    {

        var coloring = "abcdef0123456789";

        var color = "";

        for (var i = 0; i < 6; i++)

            color += coloring.charAt(Math.floor(Math.random() \* coloring.length));

    }

    Animal.findOne({name: name, value: value})

    .then(animal => {

        if (animal != null)

        {

            Animal.findOne({width: width, height: height})

            .then(animal => {

                if (animal != null)

                {

                    Animal.findOne({color: color})

                    .then(animal => {

                        if (animal != null)

                        {

                            jsonbody = animal;

                            res.redirect("./animal");

                        }

                        else

                        {

                            imgerror = {error: "Color error", name: name, value: value, width: width, height: height, color: color};

                            res.send(imgerror);

                        }

                    });

                }

                else

                {

                    imgerror = {error: "Size error", name: name, value: value, width: width, height: height};

                    res.send(imgerror);

                }

            });

        }

        else

        {

            imgerror = {error: "Name error", name: name, value: value};

            res.send(imgerror);

        }

    });

});

module.exports = router;

**animal.js**

const mongoose = require("mongoose");

const Schema = mongoose.Schema;

const AnimalSchema =  new Schema({

    name: String,

    width: Number,

    height: Number,

    color: String,

    image: String,

    value: Number

});

const Animal = mongoose.model("animal", AnimalSchema);

module.exports = Animal;

**App.js**

import React, { useState } from 'react';

function App() {

  const [name, setName] = useState('')

  const [value, setValue] = useState('')

  const [width, setWidth] = useState('')

  const [height, setHeight] = useState('')

  const [color, setColor] = useState('')

  const [image, setImage] = useState('')

  const [img, setImg] = useState('')

  /// Функция получения всех параметров

  function startSearch(event) {

    event.preventDefault()

    if (name.trim()) {

      var animalname = null;

      var animalvalue = null;

      animalname = name;

      animalvalue = value;

      let animanlsJSON = {

        name: animalname,

        value: animalvalue,

      };

      fetch('/api/findanimal', {

        method: 'POST',

        headers: {

          'Content-Type': 'application/json'

        },

        body: JSON.stringify(animanlsJSON)

        })

        .then(response => {

          if(!response.ok) {

            throw new Error(response.statusText)

          }

          return response;

        })

            .then(response => response.json())

            .then(result => {

              console.log(result)

              if (result.error !== "Image not found")

              {

                setValue(result.value)

                setWidth(result.width)

                setHeight(result.height)

                setColor(result.color)

                setImage('')

              }

              else

              setImage('Nothing in DB')

            });

    }

  }

  /// Функция получения картинки

  function sendReq(event) {

    event.preventDefault()

    if (name.trim()) {

      var animalname = null;

      var animalvalue = null;

      var imgwidth = null;

      var imgheight = null;

      var imgcolor = null;

      animalname = name;

      animalvalue = value;

      imgwidth = width;

      imgheight = height;

      imgcolor = color;

      let animanlsJSON = {

        name: animalname,

        value: animalvalue,

        width: imgwidth,

        height: imgheight,

        color: imgcolor

      };

      fetch('/api/animal', {

        method: 'POST',

        headers: {

          'Content-Type': 'application/json'

        },

        body: JSON.stringify(animanlsJSON)

        })

        .then(response => {

          if(!response.ok) {

            throw new Error(response.statusText)

          }

          return response;

        })

            .then(response => response.json())

            .then(result => {

              console.log(result)

              if (result.image != null)

              {

                setImage(result.image+'.jpg')

                setImg('../images/'+ result.image + '.jpg')

              }

              else

              setImage('Image not found')

            });

    }

  }

  return (

    <div className="wrapper">

      <label>

          <a>Название животного: </a>

          <a style={{marginLeft: 25}}>Значение: </a>

          <br></br>

          <input

            value={name}

            onChange={event => setName(event.target.value)}

          />

          <input

            style={{marginLeft: 15}}

            value={value}

            onChange={event => setValue(event.target.value)}

          /><p></p>

          <a>Ширина: </a>

          <a style={{marginLeft: 117}}>Высота: </a>

          <br></br>

          <input

            value={width}

            onChange={event => setWidth(event.target.value)}

          ></input>

          <input

            style={{marginLeft: 15}}

            value={height}

            onChange={event => setHeight(event.target.value)}

          ></input><p></p>

          <a>Цвет:</a>

          <br></br>

          <input

            value={color}

            onChange={event => setColor(event.target.value)}

          ></input><p></p>

          <a>Изображение:</a>

          <br></br>

          <input

            value={image}

            onChange={event => setImage(event.target.value)}

          ></input><p></p>

          <input

            type="submit"

            value="Начать поиск по параметрам"

            onClick={sendReq}

          />

          <p></p>

          <input

            type="submit"

            value="Получить параметры с сервера"

            onClick={startSearch}

          /><p></p>

        <img

          src={img}

          onChange={event => setImg(event.target.value)}

        />

      </label>

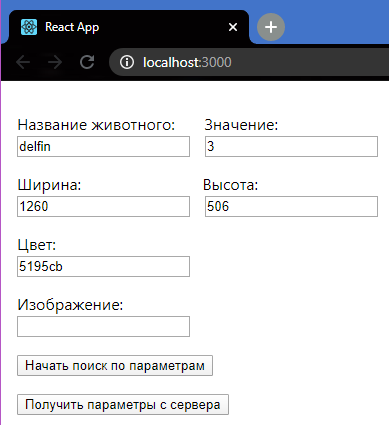
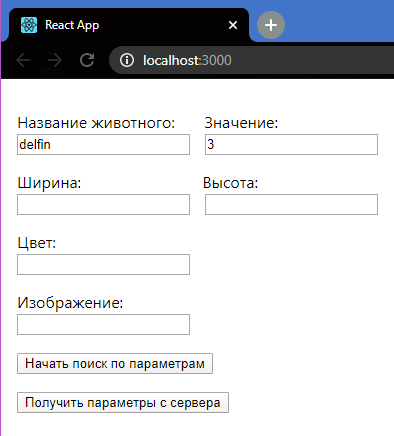
    </div>

  );

}

export default App;

**Результат работы.**

Рисунок 1 – Получение параметров с сервера

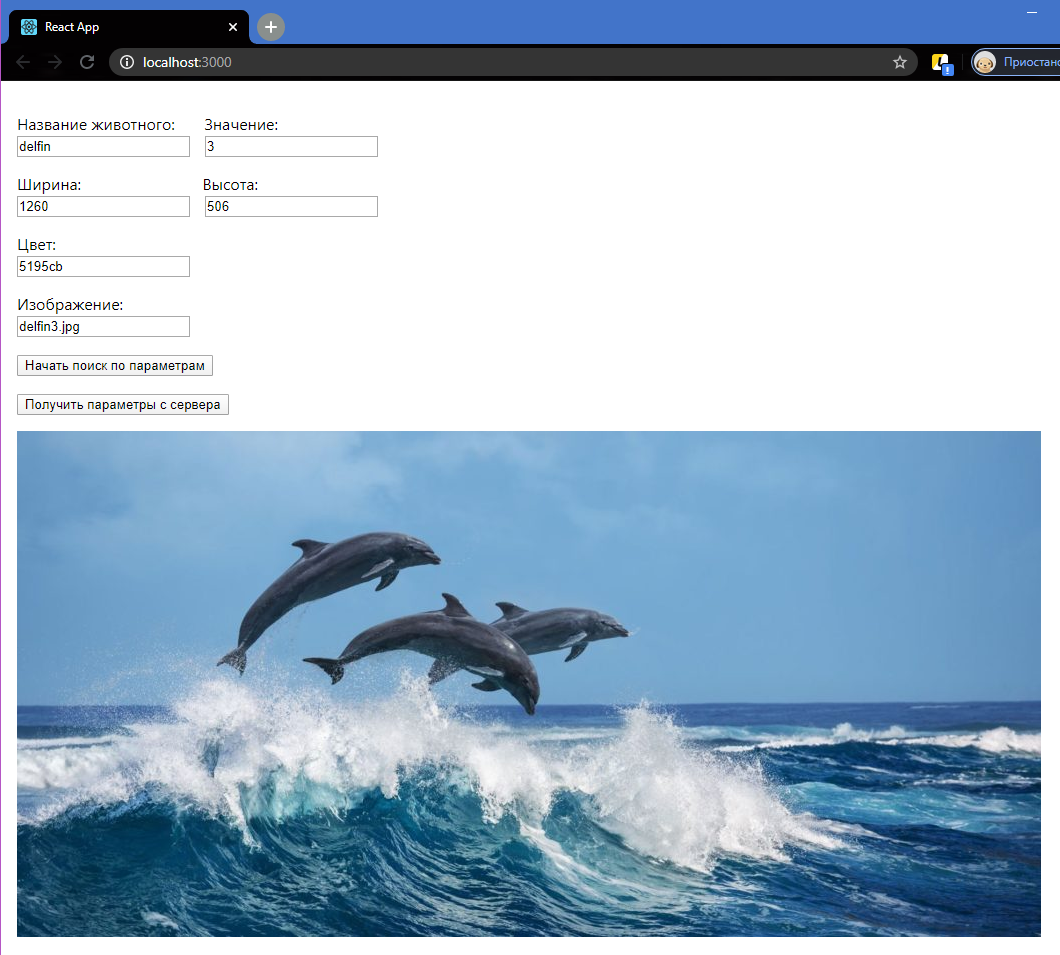


Рисунок 2 – Результат работы

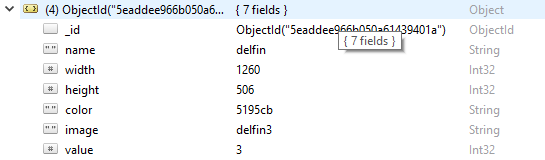


Рисунок 3 – Запись в БД

**Вывод.**

Научились реализовывать Frontend на ReactJS.