|  |  |
| --- | --- |
|  | **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 | СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ |

Отчет по лабораторной работе №8

«Разработка пользовательского интерфейса с использованием библиотеки React.»

Студент Мартынова Полина Владимировна.

*фамилия, имя, отчество*

Группа ИУ5-51Б.

Студент 12.12.2021 **Мартынова П.В.**

*подпись, дата фамилия, и.о.*

Преподаватель 12.12.2021 **Гапанюк Ю.Е.** *подпись, дата фамилия, и.о.*

2021г.

**Задание:**

На основе [методических указаний](https://github.com/iu5team/iu5web-fall-2021/blob/main/tutorials/lab8/lab8_tutorial.md) разработайте React-приложение. Для создания приложения необходимо решить следующие задачи:

1. Создать стартовый React-проект. Удалить неиспользуемый код. Организовать директории для страниц, компонентов, утилит и работы с сетью.
2. Организовать роутниг в веб-приложении.
3. Разработать базовые страницы, на которых будут отображаться сущности из выбранной вами предметной области:
   1. Стартовая страница.
   2. Страница просмотра списка объектов.
   3. Страница просмотра конкретного объекта.
4. Вынести переиспользуемые компоненты в отдельные файлы:
   1. Для навигации по приложению можно добавить header.
   2. Для отображения дополнительной информации (данные о студенте и предметной области) можно использовать footer.
   3. Источники ввода-вывода (поля ввода (inputs)/формы/текстовые блоки).
   4. Переиспользуемые таблицы/гриды.
5. Добавить асинхронные запросы в разработанный API, чтобы страница получала данные с сервера.
6. Если в Вашем проекте реализована сложная логика работы с состоянием приложения, то рекомендуется добавить пользовательские хуки.
7. Страницы приложения должны хорошо отображаться как на больших, так и на маленьких экранах.

**Код:**

*App.js:*

import {BrowserRouter, Routes as Switch, Link, Route} from "react-router-dom";

import APIPage from "./pages/APIPage";

import Characters from "./pages/characters\_page\_main/characters";

import StartPage from "./pages/StartPage";

function App() {

  return (

      <BrowserRouter basename="/" >

        <div>

          <ul>

            <li>

              <Link to="/">Старт</Link>

            </li>

            <li>

              <Link to="/list">Список</Link>

            </li>

            <li>

              <Link to="/new">Хочу на страницу с чем-то новеньким</Link>

            </li>

          </ul>

          <hr />

          <Switch>

            <Route exact path="/" element={<h1>Добро пожаловать на стартовую страницу</h1>}/>

            <Route exact path="/list" element={<StartPage/>}/>

            <Route path="/new" element={<Characters/>}/>

          </Switch>

        </div>

      </BrowserRouter>

  );

}

export default App;

*StartPage.js:*

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

const data = [

    'Москва',

    'Санкт-Петербург',

    'Владивосток',

    'Владимир',

    'Суздаль',

    'Волгоград',

    'Астрахань',

    'Коломна',

    'Ульяновск',

    'Раменское',

    'Люберцы',

]

function StartPage() {

    const [randomName, setRandomName] = useState();

    const [names, setNames] = useState(data);

    const [showNames, setShowNames] = useState(false);

    const handleShowNames = () =>{

        setShowNames(true)

    }

    const handleHideNames = () =>{

        setShowNames(false)

    }

    const handleSwitch = () =>{

        setShowNames(state => !state)

    }

    useEffect(()=>{

        console.log('Этот код выполняется только на первом рендере компонента')

        // В данном примере можно наблюдать Spread syntax (Троеточие перед массивом)

        setNames(names=>[...names, 'Бедный студент'])

        return () => {

            console.log('Этот код выполняется, когда компонент будет размонтирован')

        }

    },[])

    useEffect(()=>{

        console.log('Этот код выполняется каждый раз, когда изменится состояние showNames ')

        setRandomName(names[Math.floor(Math.random()\*names.length)])

    },[showNames])

    return (

        <div>

            <h3>Случайный город из списка: {randomName}</h3>

            <button onClick={handleShowNames}>Хочу увидеть список городов</button>

            <button onClick={handleHideNames}>Хочу скрыть список городов</button>

            <button onClick={handleSwitch}>{showNames ? 'Хочу скрыть список городов' :'Хочу увидеть список городов' }</button>

            {/\*React отрисует список только если showNames будет равен true, boolean значения игнорируются при отрисовке\*/}

            {showNames && <ul>

            {/\*Рекомендую почитать про прекрасные встроенные методы массивов в JavaScript    \*/}

            {names.map((name, index)=>{

                return (

                    <li key={index}>

                        <span>{name}</span>

                    </li>

                )

            })}

            </ul>

            }

        </div>

    );

}

export default StartPage;

*Characters.js:*

import React, {useState} from "react";

import Char\_search from "../characters\_page\_search/char\_search";

import "../../background.css"

import './characters.css'

const Characters = () => {

  function rotate\_all(event) {

    const card\_all = document.querySelector('.characters\_card\_all');

    const cardItem = card\_all.querySelector('.characters\_all');

    const HalfHeight = cardItem.offsetHeight / 2;

    const HalfWidth = cardItem.offsetWidth / 2;

    cardItem.style.transform = 'rotateX('+-(event.clientY-cardItem.getBoundingClientRect().top-HalfHeight)/7+'deg) rotateY('+(event.clientX - cardItem.getBoundingClientRect().left-HalfWidth)/7+'deg)'

  }

  function stop\_rotate\_all(event) {

    const card\_all = document.querySelector('.characters\_card\_all');

    const cardItem = card\_all.querySelector('.characters\_all');

    cardItem.style.transform = 'rotate(0)'

  }

  function rotate\_stud(event) {

    const card\_all = document.querySelector('.characters\_card\_students');

    const cardItem = card\_all.querySelector('.characters\_students');

    const HalfHeight = cardItem.offsetHeight / 2;

    const HalfWidth = cardItem.offsetWidth / 2;

    cardItem.style.transform = 'rotateX('+-(event.clientY-cardItem.getBoundingClientRect().top-HalfHeight)/7+'deg) rotateY('+(event.clientX - cardItem.getBoundingClientRect().left-HalfWidth)/7+'deg)'

  }

  function stop\_rotate\_stud(event) {

    const card\_all = document.querySelector('.characters\_card\_students');

    const cardItem = card\_all.querySelector('.characters\_students');

    cardItem.style.transform = 'rotate(0)'

  }

  function rotate\_stuff(event) {

    const card\_all = document.querySelector('.characters\_card\_stuff');

    const cardItem = card\_all.querySelector('.characters\_stuff');

    const HalfHeight = cardItem.offsetHeight / 2;

    const HalfWidth = cardItem.offsetWidth / 2;

    cardItem.style.transform = 'rotateX('+-(event.clientY-cardItem.getBoundingClientRect().top-HalfHeight)/7+'deg) rotateY('+(event.clientX - cardItem.getBoundingClientRect().left-HalfWidth)/7+'deg)'

  }

  function stop\_rotate\_stuff(event) {

    const card\_all = document.querySelector('.characters\_card\_stuff');

    const cardItem = card\_all.querySelector('.characters\_stuff');

    cardItem.style.transform = 'rotate(0)'

  }

  const [menuActive, setMenuActive] = useState(false);

  const [charSearchActive, setCharSearchActive] = useState(false);

  const items = [{value:"a", href:'./main', icon: ""},

  {value:"b", href:'./main', icon: ""},

  {value:"c", href:'./main', icon: ""},

  {value:"d", href:'./main', icon: ""}]

  var [n, setN] = useState(0) // all/characters/stuff

  function all() {

    setN(1)

    setCharSearchActive(!charSearchActive)

  }

  function stud() {

    setN(2)

    setCharSearchActive(!charSearchActive)

  }

  function stuff() {

    setN(3)

    setCharSearchActive(!charSearchActive)

  }

    return (

      <>

        <div className = "background">

          <div className = "characters\_title">Characters</div>

          <div className = "c1"></div>

          <div className = "c2"></div>

          <div className = "c3"></div>

          <div className = "c4"></div>

          <div className = "c5"></div>

          <div className = "c6"></div>

          <div className = "characters\_card\_all" onMouseMove = {(event) => rotate\_all(event)} onMouseOut = {(event) => stop\_rotate\_all(event)}>

            <div className = "characters\_all"></div>

          </div>

          <div className = "characters\_card\_stuff" onMouseMove = {(event) => rotate\_stuff(event)} onMouseOut = {(event) => stop\_rotate\_stuff(event)}>

            <div className = "characters\_stuff"></div>

          </div>

          <div className = "characters\_card\_students" onMouseMove = {(event) => rotate\_stud(event)} onMouseOut = {(event) => stop\_rotate\_stud(event)}>

            <div className = "characters\_students"></div>

          </div>

          <div className = "cont\_link\_all">

            <div className = "link" onClick = {() => all()}>

              <div className = "text\_link">All charecters</div>

            </div>

          </div>

          <div className = "cont\_link\_stud">

            <div className = "link" onClick = {() => stud()}>

              <div className = "text\_link">Hogwarts students</div>

            </div>

          </div>

          <div className = "cont\_link\_stuff">

            <div className = "link" onClick = {() => stuff()}>

              <div className = "text\_link">Hogwarts stuff</div>

            </div>

          </div>

        </div>

        <Char\_search n = {n} active = {charSearchActive} setActive = {setCharSearchActive}/>

      </>

    );

};

export default Characters;

*Char\_search.js:*

import React, {useEffect, useState} from "react";

import './char\_search.css';

import $ from 'jquery';

import Char\_info from "./character\_info";

const Char\_search = ({n, active, setActive}) => {

    var i = 1;

    $('.card-block').on("dblclick", ( function() {

        if(i%2 === 0) {

            $(this).find('.card').removeClass('flip')

            i++;

        }

        else {

            $(this).find('.card').addClass('flip');

            i--;

        }

    }))

    var [data, setData] = useState([]);

    var [char, setChar] = useState({});

    const [CharName, SetCharName] = useState("");

    const [charInfoActive, setCharInfoActive] = useState(false);

    var path\_api = "http://hp-api.herokuapp.com/api/characters"

        useEffect(() => {

        var request = new XMLHttpRequest();

        request.open('GET', path\_api);

        request.responseType = 'json';

        request.send();

        request.onload = function() {

            setData(request.response)

        }

    }, []);

    function char\_infoo(c) {

        setChar(c);

        setActive(false);

        setCharInfoActive(true);

    }

    return (

        <>

            <div className = {active ? 'main active':'main'}>

                <div className = "card-block">

                    <div className = "card">

                        <div className = "side front">

                            <div className = "cancel\_frame">

                                <div className = "cancel\_char" onClick = {() => setActive(false)}></div>

                            </div>

                            <div className = 'double\_click'></div>

                            <div className = "search">

                                    <input

                                    className="search\_input"

                                    placeholder="Characters name:"

                                    type="text"

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

                                    />

                            </div>

                            <ul className = "char\_el\_list">

                                {

                                  data

                                    .filter(char => {

                                      if(CharName === "") {

                                          return char;

                                      } else if (char.name.toLowerCase().includes(CharName.toLowerCase())) {

                                          return char;

                                      } else {

                                        return "";

                                      }

                                    })

                                    .map(char => (

                                        <li key = {char.name}>

                                          <div className = "char\_el\_name">

                                              <div className = "char\_el\_img"></div>

                                              <div className = "char\_el\_name\_text" onClick = {() => char\_infoo(char)}>{char.name}</div>

                                          </div>

                                        </li>

                                    ))

                                }

                            </ul>

                        </div>

                        <div className = {n === 1 ? 'side back one': n === 2 ? 'side back two': 'side back three'}></div>

                    </div>

                </div>

            </div>

            <Char\_info n = {n} char = {char} active = {charInfoActive} setActive = {setCharInfoActive}/>

        </>

    );

};

export default Char\_search;

*Character\_info.js:*

import React, {useState} from "react";

import './char\_search.css';

import $ from 'jquery';

import Char\_search from "./char\_search";

const Char\_info = ({n, char, active, setActive}) => {

    var i = 1;

    $('.card-block').on("dblclick", ( function() {

        if(i%2 === 0) {

            $(this).find('.card').removeClass('flip')

            i++;

        }

        else {

            $(this).find('.card').addClass('flip');

            i--;

        }

    }))

    var ImgStyle = {

        height: '25vh',

        width: '20vh',

        borderRadius: '100%',

        overflow: 'hidden',

        backgroundImage: "url(" + char.image + ")",

        backgroundSize: '100%',

        backgroundPosition: 'center'

      };

    const [charSearchActive, setCharSearchActive] = useState(false);

    // function switch\_search() {

    //     setActive(false);

    //     setCharSearchActive(true);

    // }

        return(

        <>

            <div className = {active ? 'main active1':'main'}>

                <div className = "card-block">

                    <div className = "card">

                        <div className = "side front">

                            <div className = "cancel\_frame\_info">

                                <div className = "cancel\_frame">

                                    {/\* <div className = 'back\_char' onClick = {() => switch\_search()}/> \*/}

                                    <div className = "cancel\_char" onClick = {() => setActive(false)}/>

                                </div>

                            </div>

                            <div className = 'double\_click'></div>

                            <div style = {ImgStyle}/>

                            <div className = "char\_main">

                                <div className = 'char\_str'>

                                    <div className = "char\_item\_name">Name:</div>

                                    <div className = 'char\_item\_info'>{char.name}</div>

                                </div>

                                <div className = 'char\_str'>

                                    <div className = "char\_item\_name">Species:</div>

                                    <div className = 'char\_item\_info'>{char.species}</div>

                                </div>

                                <div className = 'char\_str'>

                                    <div className = "char\_item\_name">Gender</div>

                                    <div className = 'char\_item\_info'>{char.gender}</div>

                                </div>

                                <div className = 'char\_str'>

                                    <div className = "char\_item\_name">House: </div>

                                    <div className = 'char\_item\_info'>{char.house}</div>

                                </div>

                                <div className = 'char\_str'>

                                    <div className = "char\_item\_name">Date of birth: </div>

                                    <div className = 'char\_item\_info'>{char.dateOfBirth}</div>

                                </div>

                                <div className = 'char\_str'>

                                    <div className = "char\_item\_name">Wizard: </div>

                                    <div className = 'char\_item\_info'>{String(char.wizard)}</div>

                                </div>

                                <div className = 'char\_str'>

                                    <div className = "char\_item\_name">Ancestry: </div>

                                    <div className = 'char\_item\_info'>{char.ancestry}</div>

                                </div>

                                <div className = 'char\_str'>

                                    <div className = "char\_item\_name">Eye colour: </div>

                                    <div className = 'char\_item\_info'>{char.eyeColour}</div>

                                </div>

                                <div className = 'char\_str'>

                                    <div className = "char\_item\_name">Hair colour: </div>

                                    <div className = 'char\_item\_info'>{char.hairColour}</div>

                                </div>

                                                                <div className = 'char\_str'>

                                    <div className = "char\_item\_name">Patronus: </div>

                                    <div className = 'char\_item\_info'>{char.patronus}</div>

                                </div>

                                <div className = 'char\_str'>

                                    <div className = "char\_item\_name">Actor: </div>

                                    <div className = 'char\_item\_info'>{char.actor}</div>

                                </div>

                                <div className = 'char\_str'>

                                    <div className = "char\_item\_name">Alive: </div>

                                    <div className = 'char\_item\_info'>{String(char.alive)}</div>

                                </div>

                            </div>

                        </div>

                        <div className = {n === 1 ? 'side back one': n === 2 ? 'side back two': 'side back three'}></div>

                    </div>

                </div>

            </div>

            {/\* <Char\_search n = {n} active = {charSearchActive} setActive = {setCharSearchActive}/> \*/}

        </>

    )

};

export default Char\_info;

**Демонстрация:**









