

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет приложений»

Отчет по лабораторной работе №8

«Разработка пользовательского интерфейса с использованием библиотеки
React»

Выполнил:
студент группы ИУ5-53Б
Баркалова И.В.

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.

Задание:

На основе [методических указаний](#) 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 роутинг в веб-приложении.
3. Разработать базовые страницы, на которых будут отображаться сущности из выбранной вами предметной области:
 - i. Стартовая страница.
 - ii. Страница просмотра списка объектов.
 - iii. Страница просмотра конкретного объекта.
4. Вынести переиспользуемые компоненты в отдельные файлы:
 - i. Для навигации по приложению можно добавить header.
 - ii. Для отображения дополнительной информации (данные о студенте и предметной области) можно использовать footer.
 - iii. Источники ввода-вывода (поля ввода (inputs)/формы/текстовые блоки).
 - iv. 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.

Текст программы

App.js

```
import logo from './logo.svg';
import './App.css';

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

import page0 from "./page0.js";
import Page2 from "./Page2.js";
```

```

import DP from "../DP.js";

function App() {

  return (
    <BrowserRouter basename="/" >
      <div>
        <ul>
          <li>
            <Link to="/">Старт</Link>
          </li>
          <li>
            <Link to="/new">Хочу на страницу с чем-то новеньким</Link>
          </li>
        </ul>
        <hr />
        <Switch>
          <Route exact path="/">
            <h1>Это наша стартовая страница</h1>
          </Route>
          <Route path="/new" component={Page2} />
          <Route path="/detail" component={DP} />

        </Switch>
      </div>
    </BrowserRouter>
  );
}

export default App;

```

Page2.js

```

import React from "react";
import './Page2.css';
import {Button, Card, Col, Row} from "react-bootstrap";
import Switch from "react-router-dom/es/Switch";
import Route from "react-router-dom/es/Route";
import DP from "../DP.js";
import BrowserRouter from "react-router-dom/es/BrowserRouter";
import {Link} from "react-router-dom";

class Page2 extends React.Component {
  numb;
  Open(numb) {
    localStorage.setItem("key", 8)
    window.open("/detail", "_blank")
  }

  // Constructor
  constructor(props) {
    super(props);

    this.state = {
      items: [],
      DataisLoaded: false
    };
  }
}

```

```

// ComponentDidMount is used to
// execute the code
componentDidMount() {
  fetch(
    "http://127.0.0.1:8000/os/")
    .then((res) => res.json())
    .then((json) => {
      this.setState({
        items: json,
        DataisLoaded: true
      });
    })
  })
}

render() {
  const {DataisLoaded, items} = this.state;
  if (!DataisLoaded) return <div>
    <h1> Pleses wait some time.... </h1></div>;

  return (
    <BrowserRouter basename="/new">

      <div className="App">
        <h1> Fetch data from an api in react </h1> {
          <Row xs={4} md={4} className="g-4">
            {items.map((item, index) => {
              return <Col>
                <Card key={index} className="card">
                  <Card.Img className="cardImage"
variant="top" src={item.src} height={200}
width={300}/>
                <Card.Body>
                  <div className="textStyle">

<Card.Title>{item.os_name}</Card.Title>
                  </div>
                  <div className="textStyle">
                    <Card.Text>
                      {item.os_descript}
                    </Card.Text>
                  </div>
                  <Button className="cardButton"
onClick={() => this.handleRegionClick(item.pk)}>Открыть</Button>
                </Card.Body>
              </Card>
            </Col>
          </Row>
        </div>
      </div>
    </BrowserRouter>
  );
}

handleRegionClick(id) {
  localStorage.setItem("key", id)
  window.open("/detail", "_blank")
}
}

```

```
export default Page2;
```

DP.js

```
import React from "react";
import './Page2.css';
import {Button, Card, Col, Row} from "react-bootstrap";
import Switch from "react-router-dom/es/Switch";
import Route from "react-router-dom/es/Route";
import DP from "./DP.js";
class Page2 extends React.Component {

  // Constructor
  numb;
  constructor(props) {
    super(props);

    this.state = {
      items: [],
      DataisLoaded: false
    };
  }

  // ComponentDidMount is used to
  // execute the code
  componentDidMount() {
    fetch(
      "http://127.0.0.1:8000/os/")
      .then((res) => res.json())
      .then((json) => {
        this.setState({
          items: json,
          DataisLoaded: true
        });
      })
  }

  render() {
    this.numb = localStorage.getItem("key");
    console.log("Внимание", this.numb);
    const { DataisLoaded, items } = this.state;
    if (!DataisLoaded) return <div>
      <h1> Pleses wait some time.... </h1> </div> ;

    return (
      <div className = "App">
        <h1>Детальная информация об ОС</h1> {
          <Row xs={4} md={4} className="g-4">
            {items.map((item, index)=>{
              if(item.pk==this.numb)
                return<Col >
                  <Card key={index} className="card">
                    <Card.Img className="cardImage" variant="top"
src={item.src} height={500} width={700} />
                    <Card.Body>
                      <div className="textStyle">
                        <Card.Title>{item.os_name}</Card.Title>
                      </div>
                      <div className="textStyle">
                        <Card.Text>
                          {item.os_descript}

```

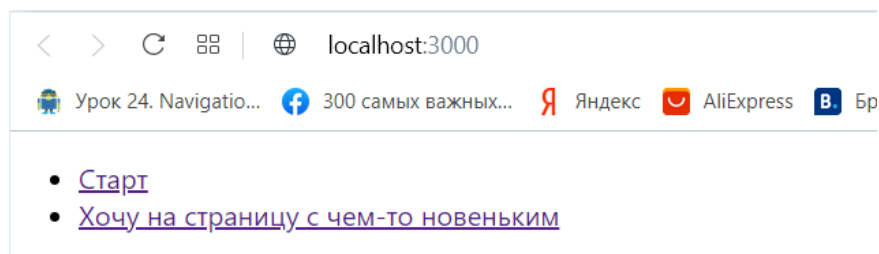
```

        </Card.Text>
      </div>
    </Card.Body>
  </Card>
</Col>
  </Row>
} } }
</div>
}
);
}
}

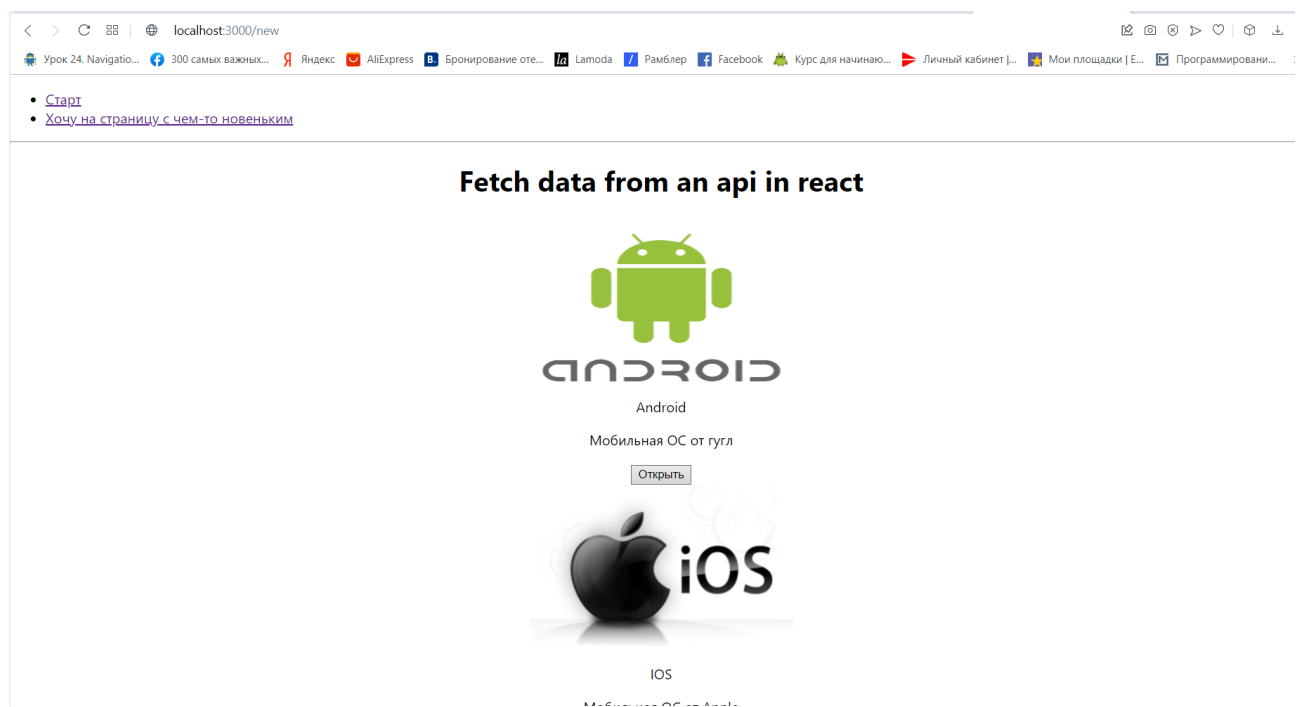
export default Page2;

```

Результат работы программы



Это наша стартовая страница



Детальная информация об ОС

