

Отчет по лабораторной работе №8 по курсу
"Разработка Интернет-Приложений"
«Разработка пользовательского интерфейса с использованием
библиотеки React.»

Выполнил:

Студент группы

ИУ5-55Б

Бахман А.А.

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:

src/App.css

```
.App {
  text-align: center;
}

.App-logo {
  height: 40vmin;
  pointer-events: none;
}

@media (prefers-reduced-motion: no-preference) {
  .App-logo {
    animation: App-logo-spin infinite 20s linear;
  }
}

.App-header {
  background-color: #282c34;
```

```

min-height: 100vh;
display: flex;
flex-direction: column;
align-items: center;
justify-content: center;
font-size: calc(10px + 2vmin);
color: white;
}

.App-link {
  color: #61dafb;
}

@keyframes App-logo-spin {
  from {
    transform: rotate(0deg);
  }
  to {
    transform: rotate(360deg);
  }
}

```

src/App.js

```

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 (
    <div>
      <BrowserRouter basename="/" >
        <div>
          <ul>
            <li>
              <Link to="/">Старт</Link>
            </li>
            <li>
              <Link to="/start">Список знаменитостей</Link>
            </li>
            <li>
              <Link to="/new">Каталог</Link>
            </li>
          </ul>
          <hr />

```

```

        <Switch>
          <Route exact path="/" >
            <h1>Это начальная страница.</h1>
            <h3>Для перемещения по страницам воспользуйтесь ссылками
сверху</h3>
          </Route>
          <Route path="/start" component={page0}/>
          <Route path="/new" component={Page2} />
          <Route path="/detail" component={DP} />
        </Switch>
      </div>
    </BrowserRouter>
  </div>
);
}

export default App;

```

src/DP.js

```

import React from "react";
import './Page2.css';
import {Button, Card, Col, Row} from "react-bootstrap";
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/app/")
      .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> Загрузка... </h1> </div> ;

return (
  <div className = "App">
    <h1>Детальная информация о Знаменитости</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.pic} alt = "картинка" width={500} height={700} />
              <Card.Body>
                <div className="textStyle">
                  <Card.Title>{item.name}</Card.Title>
                </div>
                <div className="textStyle">
                  <Card.Text>
                    Возраст:
{item.age}
                  </Card.Text>
                  <Card.Text>
                    Пол:
{item.sex==true?"мужчина":"женщина"}
                  </Card.Text>
                </div>
              </Card.Body>
            </Card>
          </Col>
        )}}
      </Row>
    </div>
  );
}
}

export default Page2;

```

src/index.css

```

body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
    sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}

```

```
code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
    monospace;
}
```

src/index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

src/page0.js

```
import React from "react";
import './App.css';
class page0 extends React.Component {

  // 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/app/")
      .then((res) => res.json())
      .then((json) => {
        this.setState({
          items: json,
          DataisLoaded: true
        });
      });
  }
}
```

```

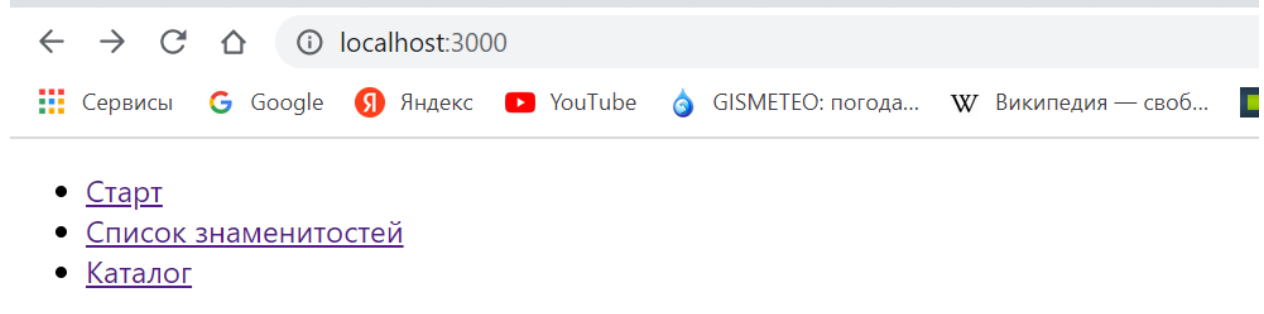
    });
  })
}
render() {
  const { DataisLoaded, items } = this.state;
  if (!DataisLoaded) return <div>
    <h1> Загрузка... </h1> </div> ;

  return (
    <div className = "App">
      <h1>Список знаменитостей</h1> {
        items.map((item) => (
          <ol key = { item.pk } >
            {item.name}
          </ol>
        ))
      }
    </div>
  );
}
}

export default page0;

```

Результат работы:



Это начальная страница.

Для перемещения по страницам воспользуйтесь ссылками сверху

- [Старт](#)
- [Список знаменитостей](#)
- [Каталог](#)

Список знаменитостей

Bauman N.E.

EINSTEIN

MOZART

SELENA GOMEZ

ANGELINA JOLIE

SHAKIRA

BEYONCE

KIRA KNITLY

- [Старт](#)
- [Список знаменитостей](#)
- [Каталог](#)

Каталог





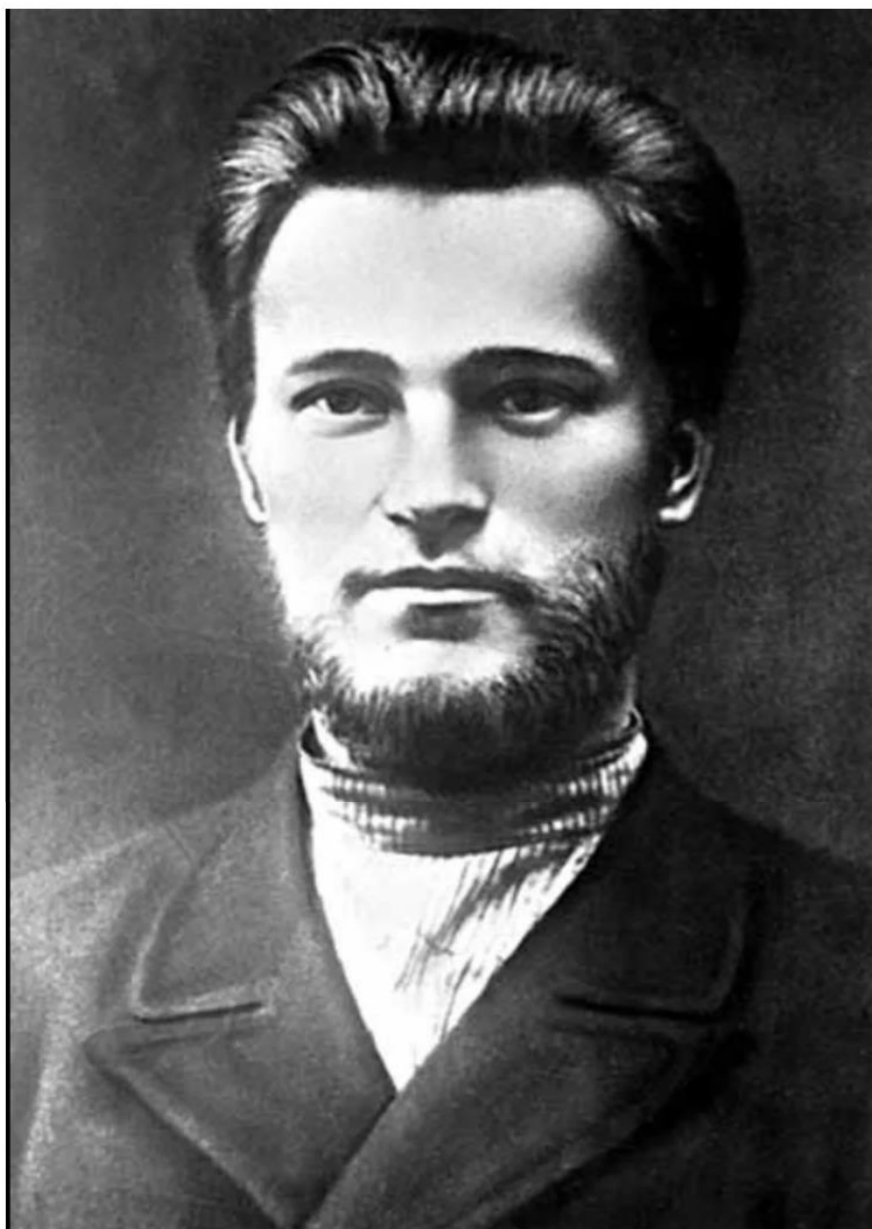
Bauman N.E.

34.00

мужчина

Открыть

Детальная информация о Знаменитости



Bauman N.E.

Возраст: 34.00

Пол: мужчина