



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»

Домашнее задание
по курсу “Разработка интернет-приложений”
“Разработка прототипа SPA приложения”

Выполнил:
студент группы ИУ5-53Б
Волгина А. Д.
29.11.21

Проверил:
Гапанюк Ю. Е.

2021 г.

Задание

Необходимо доработать ваше приложение из лабораторной работы №7 или №8. Необходимо решить следующие задачи:

1. Модифицировать базу данных:
 - База данных должна содержать основную таблицу (например, "акции") и вторую таблицу (например, "компания акции").
 - Основная таблица ссылается через внешний ключ на вторую таблицу.
2. Доработать веб-сервис:
 - Добавить в веб-сервис новый метод получения списка значений из второй таблицы.
3. Доработать фронтенд:
 - Добавить на страницу с данными основной таблицы кнопки добавления, редактирования и удаления значений через API сервиса. Для добавления и редактирования использовать страницу отображения конкретного объекта.
 - На всех страницах должны отображаться значения из таблиц вместо идентификаторов.
 - На странице отображения конкретного объекта необходимо использовать компонент `combobox` (выпадающий список). Необходимо использовать в этом компоненте новый метод веб-сервиса для второй таблицы.

Текст программы

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')  
);
```

```
reportWebVitals();
```

App.js:

```
import './App.css';
import Home from './components/Home.js'
import Search from './components/Search.js'
import React from 'react';
import { useEffect } from 'react';
import {
  BrowserRouter as Router,
  Switch,
  Route,
  Link
} from "react-router-dom";
```

```
function App() {
  const [user, setUser] = React.useState(
    {login: "no login",
    avatar_url:"no url",
    starred: "0",
    repos: [],
    followers: "0",
    following: "0",
    public_repos: "0",
    created_at: "no date",
    updated_at: "no date"
  });
  const [temp, setTemp] = React.useState(false)
  const getUsername = (event) => {
    setUser(
      {login: event.target.value,
      starred: user.starred,
      repos: user.repos,
      avatar_url: user.avatar_url,
```

```

    following: user.following,
    public_repos: user.public_repos,
    followers: user.followers,
    created_at: user.created_at,
    updated_at: user.updated_at}
  );
}
const pressEnter = (event) => {
  if (event.code.toLowerCase() === 'enter') {
    setTemp(true);
  };
}
useEffect(() => {
  let url = 'https://api.github.com/users/'
  fetch(url + user.login, {
    Authorization: `Token ${process.env.REACT_APP_API_KEY1}`
  })
  .then((response) => {
    return response.json();
  })
  .then((data) => {
    if(data.avatar_url === undefined){
      user.login = "not_found";
      return;
    }
    fetch(data.starred_url.slice(0, -15), {
      Authorization: `Token ${process.env.REACT_APP_API_KEY1}`
    })
    .then((response) => {
      return response.json();
    })
    .then((data) => {

```

```

    user.starred = data.length;

    return;
  });
  fetch(data.repos_url, {
    Authorization: `Token ${process.env.REACT_APP_API_KEY1}`
  })
    .then((response) => {
      return response.json();
    })
    .then((data) =>{
      user.repos = data;
      return;
    });
  user.public_repos = data.public_repos;
  user.avatar_url = data.avatar_url;
  user.followers = data.followers;
  user.following = data.following;
  user.created_at = data.created_at;
  user.updated_at = data.updated_at;
});
}, [user]);

return (
  <Router>
    <div className="App">
      <Link className="home" to="/" onClick={() => {setTemp(false)}}>Home</Link>
      <Switch>
        <Route path="/search">
          <Search user={user}/>
        </Route>
        <Route path="/">

```

```

      <Home onChange={getUsername} onKeyPress={pressEnter} temp={temp}/>
    </Route>
  </Switch>
</div>
</Router>
);
}

```

export default App;

GithubProfile.js:

```

import React from 'react';
import {ReactComponent as Star} from './star.svg'
import {ReactComponent as Followers} from './following-svgrepo-com.svg'

function GithubProfile({user}){
  const itemsToShow = Math.min(8, user.repos.length);
  const [ShowState, setShowState] = React.useState({
    repositories: user.repos,
    itemsToShow: itemsToShow,
    expanded: false
  });
  function showMore() {
    ShowState.itemsToShow === 8 ? (
      setShowState({ repositories: ShowState.repositories, itemsToShow: ShowState.repositories.length,
expanded: true })
    ) : (
      setShowState({ repositories: ShowState.repositories, itemsToShow: itemsToShow, expanded: false
    })
    )
  }
  const created_date = {user}.user.created_at.slice(0, 10) + " " + {user}.user.created_at.slice(11, 19);
  const updated_date = {user}.user.updated_at.slice(0, 10) + " " + {user}.user.updated_at.slice(11, 19);

```

```

return(
  <div className="main-container">
    <div className="info-block">
      <div className="avatar" style={{background: `url(${user.avatar_url}) 50% 50%`} ></div>
      <h1>{user.login}</h1>
      <button className="follow">Follow</button>
      <div className="characteristics">
        <Followers className='icon' />
        <p>{user.followers} followers</p>
        <div className="point"></div>
        <p>{user.following} following</p>
        <div className="point"></div>
        <Star className='icon' />
        <p>{user.starred}</p>
      </div>
      <p>Created at: {created_date} <br/><br/>Updated at: {updated_date}</p>
    </div>
    <div className="repo-block">
      <p>Repositories: {user.public_repos}</p>
      <div className="repo-list">
        {ShowState.repositories.slice(0, ShowState.itemsToShow).map((repa) =>
          <div className="repository">{repa.name}</div>)}
        {ShowState.repositories.length > 8 ?
          (ShowState.itemsToShow % 2 === 0 ?
            <button className="show-more" onClick={showMore}>
              {ShowState.expanded ? (
                <span>Show less</span>
              ) : (
                <span>Show more</span>
              )}
            </button>

```

```

:
<button className="show-more1" onClick={showMore}>
  {ShowState.expanded ? (
    <span>Show less</span>
  ) : (
    <span>Show more</span>
  )}
</button>) : <div></div>
}

</div>
</div>
</div>
);
}

```

```
export default GithubProfile;
```

Home.js:

```

import {
  Link,
  useHistory
} from "react-router-dom";

function Home({onChange, onKeyPress, temp}) {
  let history = useHistory();
  if({temp}.temp){
    history.push("/search");
  }
  return (
    <div className="search-panel">
      <input
        className="search-string"
        onChange={onChange}

```



```

        placeholder="Enter the GitHub nickname"
        onKeyPress={onKeyPress}
      />
      <Link to="/search">
        <button className="search-button" >Search</button>
      </Link>
    </div>
  );
}

export default Home

```

Search.js:

```

import GithubProfile from './GithubProfile.js'

function Search({user}){
  if(({user}.user.login) === 'not_found'){
    return <ShowErr/>
  }
  return <GithubProfile user={user}/>
}

function ShowErr(){
  return(
    <h1 className="error-text">User doesn't exist</h1>
  );
}

export default Search;

```

Скриншоты:

Search



LuckyHorseshoe-chan

Follow

0 followers · 0 following · ☆ 1

Created at: 2020-10-15 05:52:46

Updated at: 2022-01-18 09:34:42

Repositories: 5

[github-slideshow](#)

[IU5-Frontend-2021](#)

[iu5web-fall-2021](#)

[Labs_3sem](#)

[Labs_RIP](#)

Search

[Home](#)

User doesn't exist