**Задание по маршрутизации**

* Сделать роуты для своих лабораторных по реакту (чтобы все странички с лабами открывались каждая по своему адресу)

либо

* Сделать роуты для форм из прошлой лабораторной про [валидацию форм](https://dmitryweiner.github.io/web-lectures/React%20-%20Form%20validation.html).

/login - <LoginForm />

/registration - <RegistrationForm />

/ - <Home />

* При вводе несуществующих адресов веб-приложение переходит на /.

**Задание по запросам в сеть**

* Написать компонент, отображающий форму: ID:
* При нажатии кнопки компонент обращается по адресу <https://jsonplaceholder.typicode.com/posts/:id>.
* Также компонент обращается по адресу <https://jsonplaceholder.typicode.com/users/:userId>, где userId получен из предыдущих данных.
* Показать экране: из post - поля title и body, из user - поля name, email.

**Листинг скриптов Роуты**

**AgeCalculatorPage.js**

import React from 'react';

import AgeCalculator from '../components/AgeCalculator';

const AgeCalculatorPage = () => (

<div>

<h1>Age Calculator</h1>

<AgeCalculator />

</div>

);

export default AgeCalculatorPage;

**ProfileEditFormPage.js**

import React from 'react';

import ProfileEditForm from '../components/ProfileEditForm';

const ProfileEditFormPage = () => (

<div>

<h1>Profile Edit Form</h1>

<ProfileEditForm />

</div>

);

export default ProfileEditFormPage;

**RegistrationFormPage.js**

import React from 'react';

import RegistrationForm from '../components/RegistrationForm';

const RegistrationFormPage = () => (

<div>

<h1>Registration Form</h1>

<RegistrationForm />

</div>

);

export default RegistrationFormPage;

и ТД

**App.js**

import './App.css';

import { BrowserRouter as Router, Route, Routes, Link } from 'react-router-dom';

import HomePage from './routes/HomePage';

import CitySelectorPage from './routes/CitySelectorPage';

import CalculatorPage from './routes/CalculatorPage';

import BaseConverterPage from './routes/BaseConverterPage';

import AgeCalculatorPage from './routes/AgeCalculatorPage';

import NumberListPage from './routes/NumberListPage';

import RegistrationFormPage from './routes/RegistrationFormPage';

import ProfileEditFormPage from './routes/ProfileEditFormPage';

import React, { useState } from 'react';

const RequestTask1 = () => {

const [postId, setPostId] = useState('');

const [post, setPost] = useState(null);

const [user, setUser] = useState(null);

const [error, setError] = useState('');

const handleInputChange = (e) => {

setPostId(e.target.value);

};

const fetchPostAndUser = async () => {

try {

const postResponse = await fetch(`https://jsonplaceholder.typicode.com/posts/${postId}`);

if (!postResponse.ok) {

throw new Error('Post not found');

}

const postData = await postResponse.json();

setPost(postData);

const userResponse = await fetch(`https://jsonplaceholder.typicode.com/users/${postData.userId}`);

if (!userResponse.ok) {

throw new Error('User not found');

}

const userData = await userResponse.json();

setUser(userData);

setError('');

} catch (err) {

setError(err.message);

setPost(null);

setUser(null);

}

};

const handleSubmit = (e) => {

e.preventDefault();

fetchPostAndUser();

};

return (

<div>

<form onSubmit={handleSubmit}>

<label>

ID:

<input type="text" value={postId} onChange={handleInputChange} />

</label>

<button type="submit">Fetch Data</button>

</form>

{error && <p style={{ color: 'red' }}>{error}</p>}

{post && (

<div>

<h2>Post Details</h2>

<p><strong>Title:</strong> {post.title}</p>

<p><strong>Body:</strong> {post.body}</p>

</div>

)}

{user && (

<div>

<h2>User Details</h2>

<p><strong>Name:</strong> {user.name}</p>

<p><strong>Email:</strong> {user.email}</p>

</div>

)}

</div>

);

}

const App = () => {

return (

<div className="App">

<h1>Задание 1</h1>

<RequestTask1 />

<h1>Задание 2 - Роуты</h1>

<Router>

<nav>

<ul>

<li><Link to="/">Home</Link></li>

<li><Link to="/city-selector">City Selector</Link></li>

<li><Link to="/calculator">Calculator</Link></li>

<li><Link to="/base-converter">Base Converter</Link></li>

<li><Link to="/age-calculator">Age Calculator</Link></li>

<li><Link to="/number-list">Number List</Link></li>

<li><Link to="/registration-form">Registration Form</Link></li>

<li><Link to="/profile-edit-form">Profile Edit Form</Link></li>

</ul>

</nav>

<Routes>

<Route path="/" element={<HomePage />} />

<Route path="/city-selector" element={<CitySelectorPage />} />

<Route path="/calculator" element={<CalculatorPage />} />

<Route path="/base-converter" element={<BaseConverterPage />} />

<Route path="/age-calculator" element={<AgeCalculatorPage />} />

<Route path="/number-list" element={<NumberListPage />} />

<Route path="/registration-form" element={<RegistrationFormPage />} />

<Route path="/profile-edit-form" element={<ProfileEditFormPage />} />

<Route path="\*" element={<HomePage />} /> // Любая ссылка

</Routes>

</Router>

</div>

);

};

**Проверка корректности выполнения**







