

Задание 1

- 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 экран: из post - поля title и body, из user - поля name, email.

```
import React, { useState } from 'react';
import axios from 'axios';

const PostDetails = () => {
  const [postId, setPostId] = useState('');
  const [post, setPost] = useState(null);
  const [user, setUser] = useState(null);

  const fetchData = async () => {
    try {
      const postResponse = await axios.get(`https://jsonplaceholder.typicode.com/posts/${postId}`);
      const userResponse = await axios.get(`https://jsonplaceholder.typicode.com/users/${postResponse.data.userId}`);

      setPost(postResponse.data);
      setUser(userResponse.data);
    } catch (error) {
      console.error('Error fetching data:', error);
    }
  };

  return (
    <div>
      <form onSubmit={(e) => { e.preventDefault(); fetchData(); }}>
        <label>
          ID:
          <input type="text" value={postId} onChange={(e) => setPostId(e.target.value)} />
        </label>
        <button type="submit">Fetch Data</button>
      </form>
      {post && user && (
        <div>
          <h2>Post Details:</h2>
          <p><strong>Title:</strong> {post.title}</p>
          <p><strong>Body:</strong> {post.body}</p>
          <h2>User Details:</h2>
          <p><strong>Name:</strong> {user.name}</p>
          <p><strong>Email:</strong> {user.email}</p>
        </div>
      )}
    </div>
  );
};

export default PostDetails;
```

Задание 2

- Склонировать [репозиторий](#), запустить сервер.
- Из форм, [написанных ранее \(логин и регистрация\)](#), отправлять запросы:

- Регистрация: POST /user

```
{ "nickname": "test", "password": "123" }
```

- Логин: POST /auth

```
{ "nickname": "test", "password": "123" }
```

- После успешной регистрации перенаправлять на логин. После успешной авторизации перенаправлять на / (Home).

Перед началом

клонировать: `git clone https://github.com/dmitryweiner/mini-chat-server.git`

перейти в директорию с клоном (bash): `cd mini-chat-server`

`npm install`

`npm start`: запуск

```

import React, { useState } from 'react';
import { useHistory } from 'react-router-dom';

const Login = () => {
  const [nickname, setNickname] = useState('');
  const [password, setPassword] = useState('');
  const history = useHistory();

  const handleLogin = async () => {
    try {
      const response = await fetch('http://localhost:3000/auth', {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
        },
        body: JSON.stringify({ nickname, password }),
      });
      if (response.ok) {
        // Перенаправление на главную страницу
        history.push('/');
      } else {
        // Обработка ошибок
      }
    } catch (error) {
      console.error('Ошибка:', error);
    }
  };

  const handleRegistration = async () => {
    try {
      const response = await fetch('http://localhost:3000/user', {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
        },
        body: JSON.stringify({ nickname, password }),
      });
      if (response.ok) {
        // Перенаправление на страницу аутентификации
        history.push('/login');
      } else {
        // Обработка ошибок
      }
    } catch (error) {
      console.error('Ошибка:', error);
    }
  };
};

```

```
return (  
  <div>  
    <h2>Логин</h2>  
    <input  
      type="text"  
      placeholder="Никнейм"  
      value={nickname}  
      onChange={(e) => setNickname(e.target.value)}  
    />  
    <input  
      type="password"  
      placeholder="Пароль"  
      value={password}  
      onChange={(e) => setPassword(e.target.value)}  
    />  
    <button onClick={handleLogin}>Войти</button>  
    <h2>Регистрация</h2>  
    <button onClick={handleRegistration}>Зарегистрироваться</button>  
  </div>  
);  
};  
  
export default Login;
```

Задание 3

- 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](#).

/login - <LoginForm />

/registration - <RegistrationForm />

/ - <Home />

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

```
import React from 'react';
import { BrowserRouter as Router, Switch, Route, Redirect } from 'react-router-dom';
import LoginForm from './LoginForm';
import RegistrationForm from './RegistrationForm';
import Home from './Home';

const App = () => {
  return (
    <Router>
      <Switch>
        <Route path="/login" component={LoginForm} />
        <Route path="/registration" component={RegistrationForm} />
        <Route path="/" component={Home} />
        <Redirect to="/" /> {/* Перенаправление на главную страницу в случае ввода несуществующего адреса */}
      </Switch>
    </Router>
  );
}

export default App;
```