|  |
| --- |
| https://lh6.googleusercontent.com/QcftzNtI05T0Y6fjdSh1Rr2rt8oqZ1IvnLvbn1jLJ7CCyteVir3k-xBLv4SL1wAgWJsRhmmJSR0UW-RP63_GQenE4vVWv05BRoZTsmIcBccVTnfxwmsnNMvjg599x9SqZd8E3dkd |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования«МИРЭА - Российский технологический университет»РТУ МИРЭА |

Институт информационных технологий (ИТ)

Кафедра инструментального и прикладного программного обеспечения (ИиППО)

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №6** | |
| **по дисциплине** | |
| «Разработка клиент-серверных приложений»  на тему  **«Реализовать аутоидентификацию в React-приложении.»** | |
| Выполнил студент группы ИКБО-20-19 | Анваржонов Ж Т |
|  |  |
| Принял | ассистент, Коваленко М.А. |

Практическая работа выполнена «10» февраля 2022 г.

(подпись студента)

Зачтено «\_\_» 2022 г.

(подпись студента)

Москва 2022

**Цель работы:** создать простое React приложение.

**Задание:**

1. Создать или дополнить существующие React-приложение формой аутоидентификации пользователей с функционалом авторизации.
2. Использовать поток кода авторизации PKCE.

**Описание выполнения работы:**

Листинг 1 – листинг admin.jsx:

*function* Admin() {  
 *return* (  
 <>  
 <h1>Страница админа</h1>  
 </>  
 );  
}  
*export default* Admin;

Листинг 2 – листинг user.jsx:

*function* User() {  
 *return* (  
 <>  
 <h1>Страница обычного пользователя</h1>  
 </>  
 );  
}  
*export default* User;

Листинг 3 – листинг auth.js:

*export const* isAuthenticated = user => !!user;  
  
*export const* isAllowed = (user, rights) =>  
 rights.some(right => user.rights.includes(right));  
  
*export const* hasRole = (user, roles) =>  
 roles.some(role => user.roles.includes(role));

Листинг 4 – листинг reducers.ts:

Покажем доработанный App.js:

*import* './App.css';  
*import* Header *from* './components/Header/Header';  
*import* {BrowserRouter, Link, Route, Routes, Switch} *from* "react-router-dom";  
*import* Dialogs *from* './components/Dialog/Dialogs';  
*import* Quote *from* './components/Quote';  
*import* {hasRole} *from* "./auth";  
*import* Admin *from* "./components/allows\_users/Admin";  
*import* User *from* "./components/allows\_users/User";  
  
*export const* user = {  
 roles: ['user'],  
 rights: ['can\_view\_articles']  
};  
  
*export const* admin = {  
 roles: ['user', 'admin'],  
 rights: ['can\_view\_articles', 'can\_view\_users']  
};  
  
*const* App = ({user}) => {  
 *return* (  
 <div className="App">  
 <Header/>  
 <BrowserRouter>  
 <h2>  
 <Link to="/dialogs">Диалоги</Link>  
 <br/>  
 <Link to="/quotes">Цитаты</Link>  
 </h2>  
 <Routes>  
 <Route path="/dialogs" element={<Dialogs/>}/>  
 {hasRole(user, ['user']) && <Route path="/user" element={<Admin/>}/>}  
 {hasRole(user, ['admin']) && <Route path="/admin" element={<User/>}/>}  
 <Route exact path='/' component={Header}/>  
 <Route path="/quotes" element={<Quote/>}/>  
 </Routes>  
 </BrowserRouter>  
 </div>  
 );  
}  
  
*export default* App;

Покажем как это выглядит на странице:

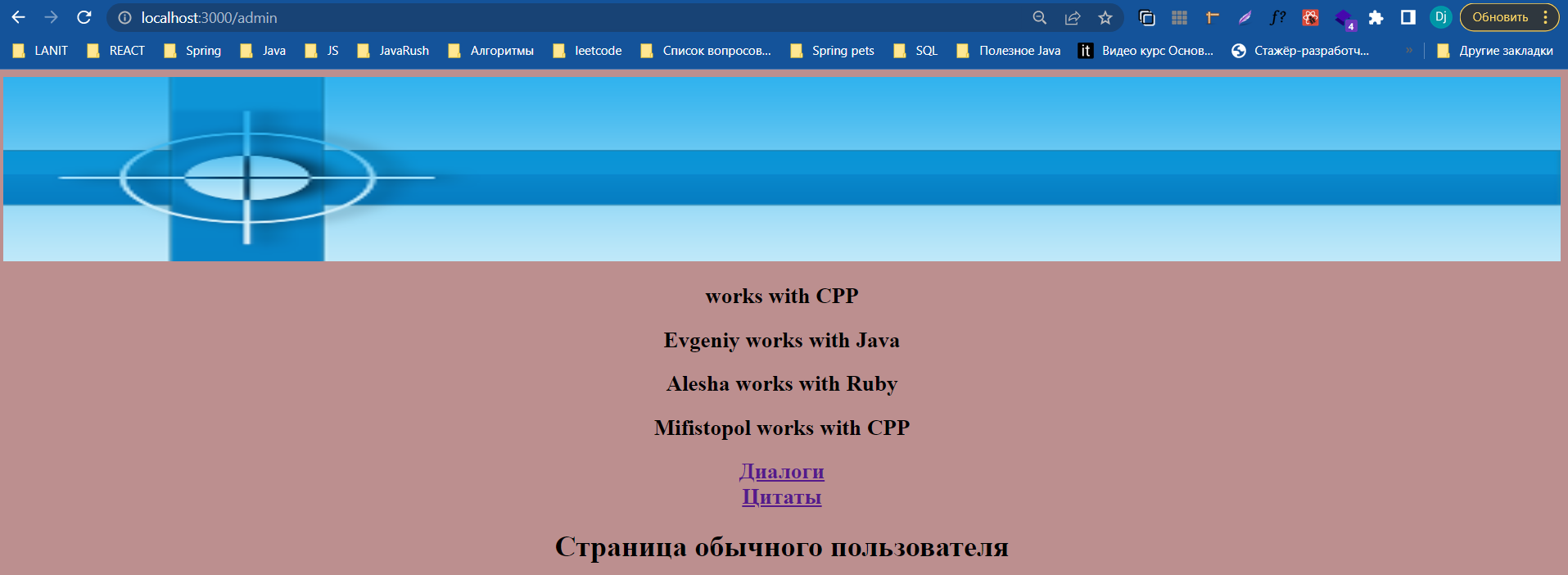


Рисунок 1- внешний вид страницы

**Вывод:** В данной практической работе был освоены и применены навыки реализации аутоидентификацию в React-приложении. В ходе выполнения работы улучшены навыки разработки React-приложения.