МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

**КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

Лабораторная работа №1

По дисциплине “Специализированные языки разметки документов”

Тема: “Авторизация и аутентификация”

Выполнил: студент группы ПО-11  
Сымоник И.А.  
Проверил:

Войцехович Г.Ю

**Цель работы:** реализовать авторизацию и аутентификацию пользователей

**Форма регистрации:**

import React, { useState } from 'react';

import axios from 'axios';

import { Link, Navigate } from 'react-router-dom';

const RegistrationForm = () => {

const [formData, setFormData] = useState({

username: '',

email: '',

password: '',

});

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

const [isRegistered, setIsRegistered] = useState(false);

const handleChange = (e) => {

const { name, value } = e.target;

setFormData({

...formData,

[name]: value,

});

};

const handleSubmit = async (e) => {

e.preventDefault();

try {

const response = await axios.post('/api/auth/registration', formData);

console.log('Ответ от сервера:', response.data);

setIsRegistered(true);

} catch (error) {

console.error('Ошибка при отправке данных на сервер:', error);

setError('Ошибка при регистрации. Попробуйте еще раз.');

}

};

if (isRegistered) {

return <Navigate to="/main" />;

}

return (

<div className="container\_form">

<form className="form" onSubmit={handleSubmit}>

<div className="form-group">

<label className="form-label" htmlFor="username">Имя пользователя:</label>

<input

type="text"

id="username"

name="username"

value={formData.username}

onChange={handleChange}

className="form-input"

required

/>

</div>

<div className="form-group">

<label className="form-label" htmlFor="email">Email:</label>

<input

type="email"

id="email"

name="email"

value={formData.email}

onChange={handleChange}

className="form-input"

required

/>

</div>

<div className="form-group">

<label className="form-label" htmlFor="password">Пароль:</label>

<input

type="password"

id="password"

name="password"

value={formData.password}

onChange={handleChange}

className="form-input"

required

/>

</div>

{error && <p className="error-message">{error}</p>}

<button type="submit" className="form-submit">Зарегистрироваться</button>

<p>У вас есть аккаунт? <Link to="/login">Войти</Link></p>

</form>

</div>

);

};

export default RegistrationForm;

**Форма входа:**

import React, { useState, useContext } from 'react';

import axios from 'axios';

import { Link, useNavigate } from 'react-router-dom';

import { AuthContext } from '../contexts/AuthContext';

const LoginForm = ({ onToggleForm }) => {

const [formData, setFormData] = useState({

username: '',

password: '',

});

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

const navigate = useNavigate();

const { handleSuccessfulLogin } = useContext(AuthContext);

const handleChange = (e) => {

const { name, value } = e.target;

setFormData({

...formData,

[name]: value,

});

};

const handleSubmit = async (e) => {

e.preventDefault();

try {

const response = await axios.post('/api/auth/login', formData);

console.log('Ответ от сервера:', response.data);

const { token, userId, isAdmin } = response.data;

if (!response.data) {

console.error('Данные от сервера не получены');

setError('Ошибка при входе. Попробуйте еще раз.');

return;

}

handleSuccessfulLogin(token, userId, isAdmin);

if (isAdmin) {

navigate('/admin');

} else {

navigate('/main');

}

} catch (error) {

console.error('Ошибка при отправке данных на сервер:', error);

setError('Ошибка при входе. Проверьте правильность введенных данных.');

}

};

return (

<div className="container\_form">

<form className="form" onSubmit={handleSubmit}>

<div className="form-group">

<label className="form-label" htmlFor="username">Имя пользователя:</label>

<input

type="text"

id="username"

name="username"

value={formData.username}

onChange={handleChange}

className="form-input"

required

/>

</div>

<div className="form-group">

<label className="form-label" htmlFor="password">Пароль:</label>

<input

type="password"

id="password"

name="password"

value={formData.password}

onChange={handleChange}

className="form-input"

required

/>

</div>

{error && <p className="error-message">{error}</p>}

<button type="submit" className="form-submit">Войти</button>

</form>

<p>У вас нет аккаунта? <Link to="/registration">Зарегистрироваться</Link></p>

</div>

);

};

export default LoginForm;

**Обработка запросов на вход:**

exports.loginUser = async (req, res) => {

try {

const { username, password } = req.body;

console.log('Попытка авторизации с данными:', { username, password });

const connection = await pool.getConnection();

const [rows] = await connection.query('SELECT \* FROM users WHERE username = ? AND password = ?', [username, password]);

connection.release();

if (username === 'admin' && password === 'admin') {

console.log('Успешная авторизация для администратора:', username);

const token = jwt.sign({ isAdmin: true }, process.env.JWT\_SECRET, { expiresIn: '1d' });

res.status(201).json({ message: 'Успешная авторизация для администратора', isAdmin: true, token });

} else if (rows.length > 0) {

console.log('Успешная авторизация для пользователя:', username);

const userId = rows[0].id;

console.log('ID пользователя:', userId);

const token = jwt.sign({ userId }, process.env.JWT\_SECRET, { expiresIn: '1d' });

res.json({ message: 'Успешная авторизация', token, userId });

} else {

console.log('Ошибка авторизации: неправильное имя пользователя или пароль');

res.status(401).json({ error: 'Неправильное имя пользователя или пароль' });

}

} catch (error) {

console.error('Ошибка при авторизации:', error);

res.status(500).json({ error: 'Ошибка при авторизации пользователя' });

}

};

**Обработка запросов на регистрацию:**

exports.registerUser = async (req, res) => {

try {

const { username, email, password } = req.body;

const connection = await pool.getConnection();

const [existingUser] = await connection.query('SELECT \* FROM users WHERE email = ?', [email]);

if (existingUser.length > 0) {

return res.status(400).json({ error: 'Пользователь с таким email уже существует' });

}

await connection.query('INSERT INTO users (username, email, password) VALUES (?, ?, ?)', [username, email, password]);

const [user] = await connection.query('SELECT id FROM users WHERE email = ?', [email]);

const userId = user[0].id;

console.log('JWT\_SECRET:', process.env.JWT\_SECRET);

const token = jwt.sign({ userId }, process.env.JWT\_SECRET, { expiresIn: '1d' });

connection.release();

res.status(201).json({ message: 'Пользователь успешно зарегистрирован', token });

} catch (error) {

console.error('Ошибка при регистрации:', error);

res.status(500).json({ error: 'Ошибка при регистрации пользователя' });

}

};

Авторизация и аутентификация в проекте работают посредством jwt токенов. Информация о пользователях хранится в mysql базе данных, которая содержит такие поля как: id пользователя, его email, jwt токен и роль пользователя.