**SERVER SIDE**

// server.js

const express = require('express');

const bodyParser = require('body-parser');

const bcrypt = require('bcrypt');

const jwt = require('jsonwebtoken');

const app = express();

const PORT = 3001;

app.use(bodyParser.json());

const users = [];

const verifyToken = (req, res, next) => {

const token = req.headers['authorization'];

if (!token) return res.sendStatus(401);

jwt.verify(token, 'secret', (err, user) => {

if (err) return res.sendStatus(403);

req.user = user;

next();

});

};

app.post('/register', async (req, res) => {

try {

const hashedPassword = await bcrypt.hash(req.body.password, 10);

const user = { username: req.body.username, password: hashedPassword };

users.push(user);

res.status(201).send('User registered successfully');

} catch {

res.status(500).send('Failed to register user');

}

});

app.post('/login', async (req, res) => {

const user = users.find(u => u.username === req.body.username);

if (user == null) {

return res.status(400).send('User not found');

}

try {

if (await bcrypt.compare(req.body.password, user.password)) {

const accessToken = jwt.sign({ username: user.username }, 'secret');

res.json({ accessToken: accessToken });

} else {

res.status(401).send('Incorrect password');

}

} catch {

res.status(500).send('Internal server error');

}

});

app.listen(PORT, () => console.log(Server running on port ${PORT}));

**CLIENT SIDE**

// App.js

import React, { useState } from 'react';

import axios from 'axios';

function App() {

const [username, setUsername] = useState('');

const [password, setPassword] = useState('');

const [message, setMessage] = useState('');

const handleRegister = async () => {

try {

await axios.post('http://localhost:3001/register', { username, password });

setMessage('User registered successfully');

} catch (error) {

console.error(error);

setMessage('Failed to register user');}};

const handleLogin = async () => {

try {

const response = await axios.post('http://localhost:3001/login', { username, password });

const accessToken = response.data.accessToken;

localStorage.setItem('accessToken', accessToken);

setMessage('Login successful');

} catch (error) {

console.error(error);

setMessage('Login failed'); }};

return (

<div>

<h1>User Authentication</h1>

<input type="text" placeholder="Username" value={username} onChange={e => setUsername(e.target.value)} />

<input type="password" placeholder="Password" value={password} onChange={e => setPassword(e.target.value)} />

<button onClick={handleRegister}>Register</button>

<button onClick={handleLogin}>Login</button>

<p>{message}</p>

</div>

);}

export default App;