issuingauthority.jsx

import React, { useState } from 'react';

const IssuingAuthority = () => {

const [documentType, setDocumentType] = useState('');

const [file, setFile] = useState(null);

const handleSubmit = (e) => {

e.preventDefault();

if (!file) return;

const reader = new FileReader();

reader.onload = (event) => {

const content = event.target.result;

const documentId = Math.random().toString(36).substr(2, 9);

const document = {

id: documentId,

type: documentType,

content: content,

issueDate: new Date().toISOString()

};

// Simulate blockchain storage using localStorage

const blockchain = JSON.parse(localStorage.getItem('blockchain') || '[]');

blockchain.push(document);

localStorage.setItem('blockchain', JSON.stringify(blockchain));

alert(`Document stored with ID: ${documentId}`);

setDocumentType('');

setFile(null);

};

reader.readAsDataURL(file);

};

return (

<div>

<h1>Issuing Authority Portal</h1>

<form onSubmit={handleSubmit}>

<div>

<label htmlFor="documentType">Document Type:</label>

<select

id="documentType"

value={documentType}

onChange={(e) => setDocumentType(e.target.value)}

required

>

<option value="">Select a document type</option>

<option value="birthCertificate">Birth Certificate</option>

<option value="academicTranscript">Academic Transcript</option>

<option value="experienceCertificate">Experience Certificate</option>

</select>

</div>

<div>

<label htmlFor="file">Upload Document:</label>

<input

type="file"

id="file"

onChange={(e) => setFile(e.target.files[0] || null)}

required

/>

</div>

<button type="submit">Generate and Store Document</button>

</form>

</div>

);

};

export default IssuingAuthority;