

**BLOCKCHAIN-BASED E-VAULT FOR LEGAL RECORDS**

Venkatarathnam Korukonda ⁽¹⁾, Muram Akhil Aravind Reddy ⁽²⁾, Chimaladinne Chinna Babu ⁽³⁾, Ediga Venkata Sivashankar Goud ⁽⁴⁾, Gudiya Tejeswararao ⁽⁵⁾, Leburi Akash ⁽⁶⁾

¹ Asst.Professor,CSE(Artificial Intelligence) Department,ABRCET,Kanigiri, Andhra Pradesh, India.

^{2,3,4,5,6} B.Tech Student, CSE(Artificial Intelligence) Department ABRCET, Kanigiri, Andhra Pradesh, India.

ABSTRACT:

In the digital age, the management and security of legal records pose significant challenges for law firms, government agencies, and individuals. Traditional paper-based record-keeping systems are prone to loss, tampering, and inefficiencies. To address these challenges, this paper proposes a blockchain-based e-vault system for the secure storage, management, and verification of legal records. The proposed system leverages blockchain technology to create a decentralized and tamper-proof repository for legal documents, such as contracts, deeds, wills, and court records. Each document is encrypted, timestamped, and stored as a transaction on the blockchain, ensuring its integrity and immutability. Smart contracts are utilized to automate the management of access permissions, document sharing, and verification processes, enhancing efficiency and transparency. Furthermore, the e-vault system incorporates advanced encryption and authentication mechanisms to protect sensitive information and ensure data privacy. Users can securely access and manage their legal records through a user-friendly interface, while maintaining control over their data sovereignty.

1.0 INTRODUCTION :

In the digital era, the management and security of legal records are paramount for ensuring transparency, integrity, and accessibility in legal proceedings. However, traditional paper-based record-keeping systems are increasingly becoming outdated, posing significant challenges for law firms, government agencies, and individuals alike. These challenges include the risk of loss, tampering, inefficiencies in document retrieval, and difficulties in maintaining data privacy and security. To address these issues, there is a growing need for innovative solutions that leverage emerging technologies to modernize record management practices in the legal domain. Blockchain technology, known for its decentralized, immutable, and transparent nature, offers a promising avenue for transforming how legal records are stored, managed, and verified. In this context, this paper proposes a blockchain-based e-vault system for the secure storage, management, and verification of legal records. By harnessing the capabilities of blockchain technology, the proposed system aims to overcome the limitations of traditional paper-based record-keeping systems and deliver a secure, transparent, and efficient platform for managing legal documents.



2.0 Literature Review:

1. **Blockchain Technology in Legal Record Management:** Several studies have explored the potential applications of blockchain technology in the legal domain, particularly in record management. Research by Casey et al. (2018) and O'Dwyer et al. (2020) discuss how blockchain can provide a secure and transparent platform for storing, managing, and verifying legal records, reducing the risk of tampering and ensuring data integrity.

2. **Smart Contracts in Legal Operations:** Smart contracts, a key feature of blockchain technology, have been studied for their potential applications in legal operations. Research by De Filippi and Wright (2018) and Bähring (2019) examines how smart contracts can automate legal processes, such as contract execution, dispute resolution, and compliance management, leading to increased efficiency and transparency in legal transactions.

3. **Decentralized Identity Management:** Identity management is a crucial aspect of legal record management, and blockchain technology offers solutions for decentralized identity verification. Studies by Hardjono and Smith (2018) and Yu et al. (2019) explore the use of blockchain-based identity management systems to securely verify the identity of individuals and entities involved in legal transactions, enhancing trust and security.

4. **Data Privacy and Confidentiality:** Protecting the privacy and confidentiality of legal records is paramount, especially in sensitive legal proceedings. Research by Mödersheim et al. (2017) and Kosba et al. (2016) investigates privacy-preserving techniques, such as zero-knowledge proofs and cryptographic encryption, to safeguard sensitive information stored on the blockchain, ensuring compliance with data protection regulations.

5. **Regulatory and Legal Implications:** The adoption of blockchain technology in the legal domain raises regulatory and legal considerations. Studies by Werbach (2018) and Auerbach et al. (2019) examine the regulatory challenges and legal implications of blockchain-based record management systems, highlighting the need for clear guidelines and frameworks to govern their use in legal proceedings.

Overall, the existing literature provides valuable insights into the potential benefits, challenges, and implications of using blockchain technology for legal record management.

3.0 EXISTING SYSTEM :

In existing record management practices within the legal domain, reliance on traditional paper-based systems persists despite the digital advancements of recent years. Law firms, government agencies, and individuals often encounter challenges associated with the cumbersome nature of paper records, including the risk of loss, tampering, and difficulties in document retrieval. Moreover, paper-based systems lack the transparency, efficiency, and security afforded by digital technologies, leading to inefficiencies in record management processes. Centralized databases and document management systems, while offering some improvements over paper-based methods, still face issues related to data integrity, privacy, and security. These challenges underscore the need for innovative solutions that can modernize legal record management practices and address the shortcomings of existing systems.

4.0 Proposed Methodology

In real world many types of Legal documents are exists and government are managing all this documents in a single centralized server. This servers will be managed by Admin and can be bribe to alter any legal document and there will be no direct way to detect such alteration. Another most important issue is cyber-attack where attackers can hack centralized server and may crash or steal data and in such situations all data will be lost.

To overcome from above issue we are planning to migrate legal or criminal documents management to Blockchain technology which has inbuilt support for data security, verification and decentralized storage.

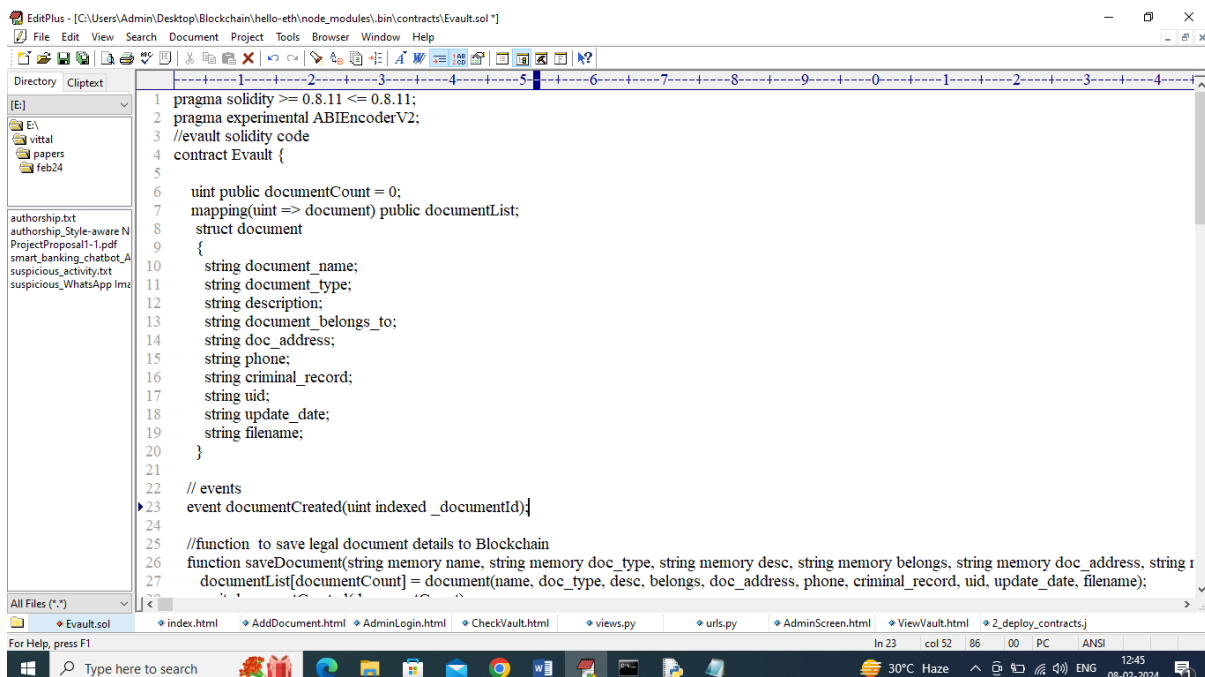
Decentralized storage: Blockchain store each records in multiple nodes, so if one node down then services can be access from other working nodes

Data Security: each block stored in Blockchain is internally encrypted so data will be secured

Data Verification: Blockchain store each data as transaction/block and associate each block with unique hash code, while storing new record Blockchain will verify hash code of all previous blocks, if data not tamper then it will result into same hash code and verification get successful otherwise data tamper can be detected.

So by employing Blockchain technology we can provide all types of securities to Legal documents which cannot be fulfilled by existing tradition single centralized servers.

To manage data with Blockchain we need to design Smart Contract using Solidity programming which contains set of functions to store and get data from Blockchain. In below screen showing Smart contract code designed to manage legal documents.



```
1 pragma solidity >= 0.8.11 <= 0.8.11;
2 pragma experimental ABIEncoderV2;
3 //evault solidity code
4 contract Evault {
5
6     uint public documentCount = 0;
7     mapping(uint => document) public documentList;
8     struct document
9     {
10        string document_name;
11        string document_type;
12        string description;
13        string document_belongs_to;
14        string doc_address;
15        string phone;
16        string criminal_record;
17        string uid;
18        string update_date;
19        string filename;
20    }
21
22    // events
23    event documentCreated(uint indexed _documentId)
24
25    //function to save legal document details to Blockchain
26    function saveDocument(string memory name, string memory doc_type, string memory desc, string memory belongs, string memory doc_address, string
27        documentList[documentCount] = document(name, doc_type, desc, belongs, doc_address, phone, criminal_record, uid, update_date, filename);
```

```

EditPlus - [C:\Users\Admin\Desktop\Blockchain\hello-eth\node_modules\bin\contracts\Evault.sol (*)
File Edit View Search Document Project Tools Browser Window Help
Directory Cliptext
[E:]
  EA
  vittal
  papers
  feb24
authorship.txt
authorship_style-aware N
ProjectProposall-1.pdf
smart_banking_chatbot_A
suspicious_activity.txt
suspicious_WhatsApp ima
All Files (*.*)
  Evault.sol
  index.html
  AddDocument.html
  AdminLogin.html
  CheckVault.html
  views.py
  urls.py
  AdminScreen.html
  ViewVault.html
  2_deploy_contracts.j
For Help, press F1
Type here to search
30°C Haze
12:45
08-02-2024
event documentCreated(uint indexed _documentId);
23
24
25 //function to save legal document details to Blockchain
26 function saveDocument(string memory name, string memory doc_type, string memory desc, string memory belongs, string memory doc_address, string i
27 documentList[documentCount] = document(name, doc_type, desc, belongs, doc_address, phone, criminal_record, uid, update_date, filename);
28 emit documentCreated(documentCount);
29 documentCount++;
30 }
31
32 //get document count
33 function getDocumentCount() public view returns (uint) {
34 return documentCount;
35 }
36
37 function getName(uint i) public view returns (string memory) {
38 document memory doc = documentList[i];
39 return doc.document_name;
40 }
41
42 function getType(uint i) public view returns (string memory) {
43 document memory doc = documentList[i];
44 return doc.document_type;
45 }
46
47 function getDesc(uint i) public view returns (string memory) {
48 document memory doc = documentList[i];
49 return doc.description;

```

In above solidity code we define all required functions need to manage legal document values. Now we need to deploy above contract in Blockchain Ethereum tool to save and get data with security. To deploy contract we need to follow below steps

1) First go inside 'hello-eth/node-modules/bin' folder and then double click on 'runBlockchain.bat' file to get below screen

```

Select C:\Windows\system32\cmd.exe
C:\Users\Admin\Desktop\Blockchain\hello-eth\node_modules\bin\truffle develop
Truffle Develop started at http://127.0.0.1:9545/

Accounts:
(0) 0xbd5c512bc386f7db107b33b4fd681943433b33f9
(1) 0x280c32f28917977dea9a2d2c15cb153014a48e7
(2) 0xef81ba08ebb90bbd3a2793baae529f55e6c5e84f
(3) 0x7e3e476918e2791d76021058015dea0271b53c
(4) 0x28c5d8c9403f8e591a8c59be8059718565f17809
(5) 0xe980550f5f6997ffa3e36f9723cc7573bf3a346fbc
(6) 0x346b3abaac3bfddbd1d01d1172f09a499ee9ad
(7) 0xa314d487a0bc5f4880b0bec0656ee47953237fc
(8) 0x75edc32f4c4fb8a71c03b084f88535ffa29388b2
(9) 0x26c15a13cc42a7f8a01b7f9137d250112bf7689

Private Keys:
(0) 818bb47e2915346a36f79f0bec2c5307565ac77bb6359a218f7b27ba932e86a
(1) 34ae98b4a4de64b59bde65b8cbced7e143e55fb7db72933c796921150d8c890
(2) 1c3cbfe9d01d939784f4d0b2a879e3e22056cfc3a096a7fb1def645915f3b59
(3) b7f8e093adc0b284431c0f0bec65f4db0423d2c4f7fcea1db1618200c9d
(4) d5974373139909844b7f7dc55e49295244abf9a9804db2f35ee636ae0916
(5) cf07cfff51a61e25e1f38d4a0ef2699f7c15c32e5fd288812a356e63d7a0d
(6) 7f5019bb93e6d4d95021c3932e89c45e4e318a52bf483f3f3e6492b1ec2837
(7) 20c0cb046aa16300218a6509a1b63a0d07aa2834f0cc1114f9dc8a7e439516
(8) fcb8a3b3711e01f09e1b997631f71327d68f0f4c7ba02925224aa0e379d
(9) daee6a412960a19cfd45efA0d67a80e92f6e0df443c7ffbc7498202f99af920

Mnemonic: announce capital blade pride sunset cannon soap thrive boy satisfy heart ordinary
Important : This mnemonic was created for you by Truffle. It is not secure.
Ensure you do not use it on production blockchains, or else you risk losing funds.

truffle(develop)> migrate

Compiling your contracts...
-----
> Compiling .\contracts\Evault.sol
ParserError: Expected identifier but got 'address'
--> project:/contracts/Evault.sol:14:15:
14 |         string address;
    |               ^^^^^^^

```

2) In above screen Ethereum started with default account and private keys and now type command as 'migrate' and press enter key to get below page

```
Select C:\Windows\system32\cmd.exe
> Network name: 'develop'
> Network id: 5777
> Block gas limit: 6721975 (0x6691b7)

2_deploy_contracts.js
-----
Deploying 'Evault'
-----
> transaction hash: 0x787bb5e8a800ba5c6105c687f3424012b39431159b521dbb7fd4e7652e4fdcea
> Blocks: 0 Seconds: 0
> contract address: 0xd374Cb05bd6187D6cF905D7bBD85f2b704fBDD29
-----
> BLOCK number: 1
> block timestamp: 1707375489
> account: 0xbDC5a12bc386F7DB107b33b4fD681943433b33f9
> balance: 99.991379684
> gas used: 4310158 (0x41c48e)
> gas price: 2 gwei
> value sent: 0 ETH
> total cost: 0.008620316 ETH

> Saving artifacts
-----
> Total cost: 0.008620316 ETH

Summary
=====
> Total deployments: 1
> Final cost: 0.008620316 ETH

- Blocks: 0 Seconds: 0

truffle(develop)>
```

- 4)
- 5) In above screen in white colour text can see 'E-Vault' contract deployed in Ethereum and got contract address also. This address we need to specify in python programming to call contract to save and get data. In below screen showing python code calling above contract

```
views.py - E:\takeoff\feb24\Evault\VaultApp\views.py (3.7.2)
File Edit Format Run Options Window Help
from django.shortcuts import render
from django.template import RequestContext
from django.contrib import messages
import json
from web3 import Web3, HTTPProvider
from django.http import HttpResponse
from django.core.files.storage import FileSystemStorage
import os
import random
from datetime import datetime

global uname, email, contract, web3, document_list
#function to call contract
def getContract():
    global contract, web3
    blockchain_address = 'http://127.0.0.1:9545'
    web3 = Web3(HTTPProvider(blockchain_address))
    web3.eth.defaultAccount = web3.eth.accounts[0]
    compiled_contract_path = 'Evault.json' #evault contract file
    deployed_contract_address = '0xd374Cb05bd6187D6cF905D7bBD85f2b704fBDD29' #contract address
    with open(compiled_contract_path) as file:
        contract_json = json.load(file) # load contract info as JSON
        contract_abi = contract_json['abi'] # fetch contract's abi - necessary to call its functions
    file.close()
    contract = web3.eth.contract(address=deployed_contract_address, abi=contract_abi)
    getContract()

def readDoc():
    global document_list, contract
    document_list = []
    count = contract.functions.getDocumentCount().call()
    for i in range(0, count):
        doc_name = contract.functions.getName(i).call()
        doc_type = contract.functions.getType(i).call()
        desc = contract.functions.getDesc(i).call()
        belongs = contract.functions.getBelongs(i).call()
        address = contract.functions.getAddress(i).call()
        phone = contract.functions.getPhone(i).call()
        criminal = contract.functions.getCriminalRecord(i).call()
        uid = contract.functions.getUid(i).call()
        today_date = contract.functions.getDate(i).call()
        # ...
```

- 6)
- 7) In above screen read red colour comments to know about contract calling in python.

5.0 RESULTS

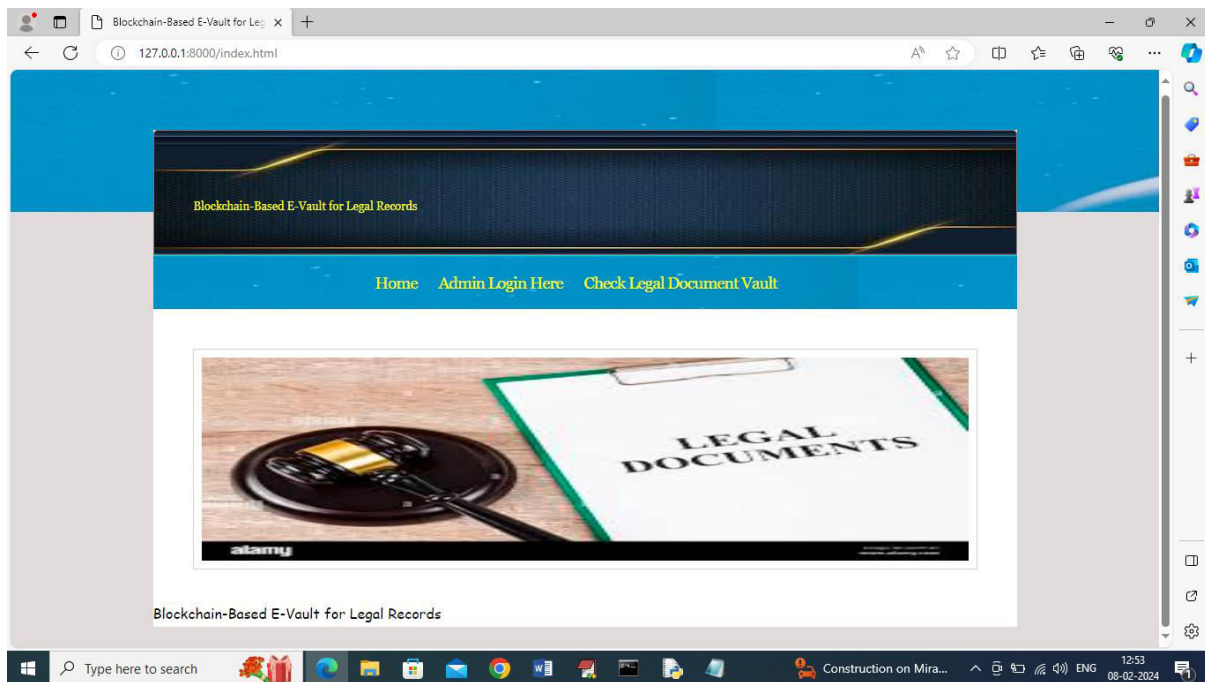
In above screens contract deployed and running and now double click on 'run.bat' file to start python server and get below page

```
C:\Windows\system32\cmd.exe
E:\takeoff\feb24\Evault>python manage.py runserver
Performing system checks...

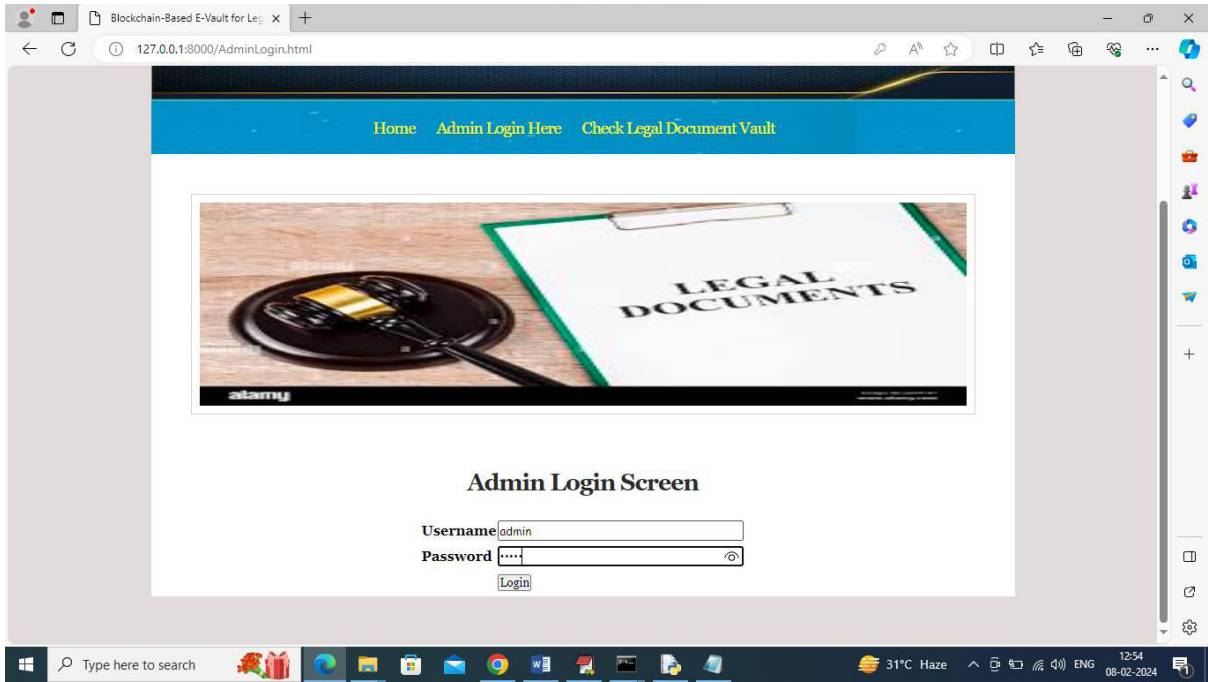
System check identified no issues (0 silenced).

You have 15 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin,
auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
February 08, 2024 - 12:51:27
Django version 2.1.7, using settings 'Vault.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

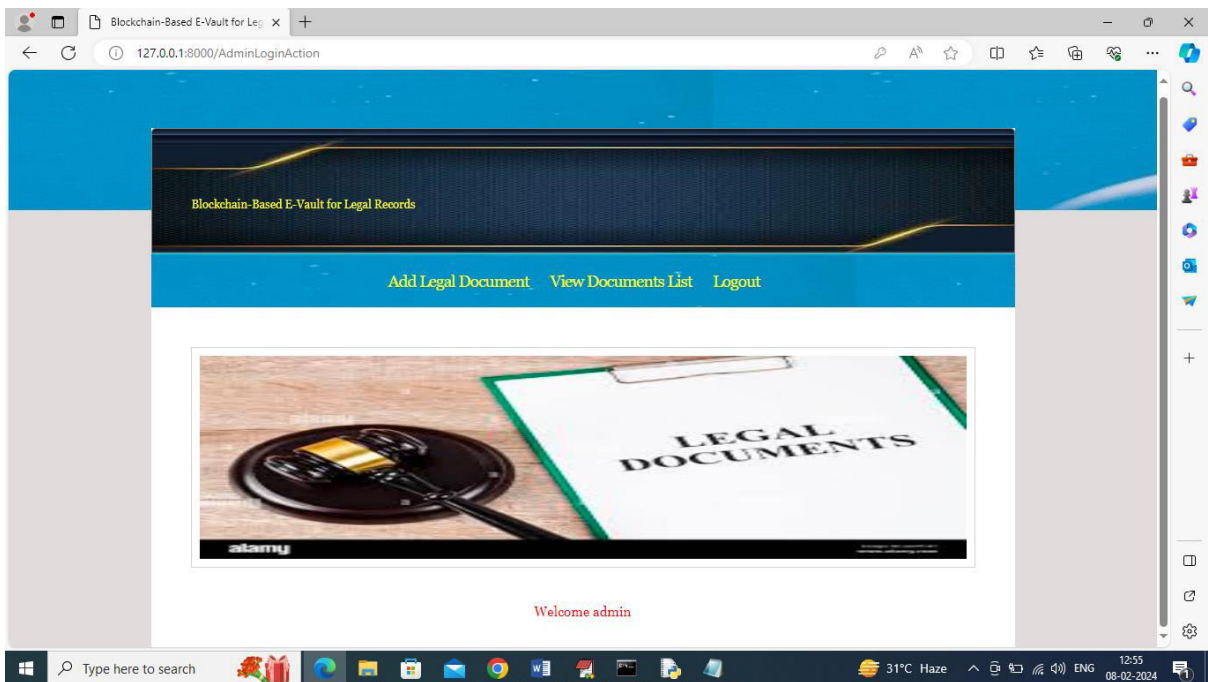
In above screen python server started and now open browser and enter URL as <http://127.0.0.1:8000/index.html> and then press enter key to get below page



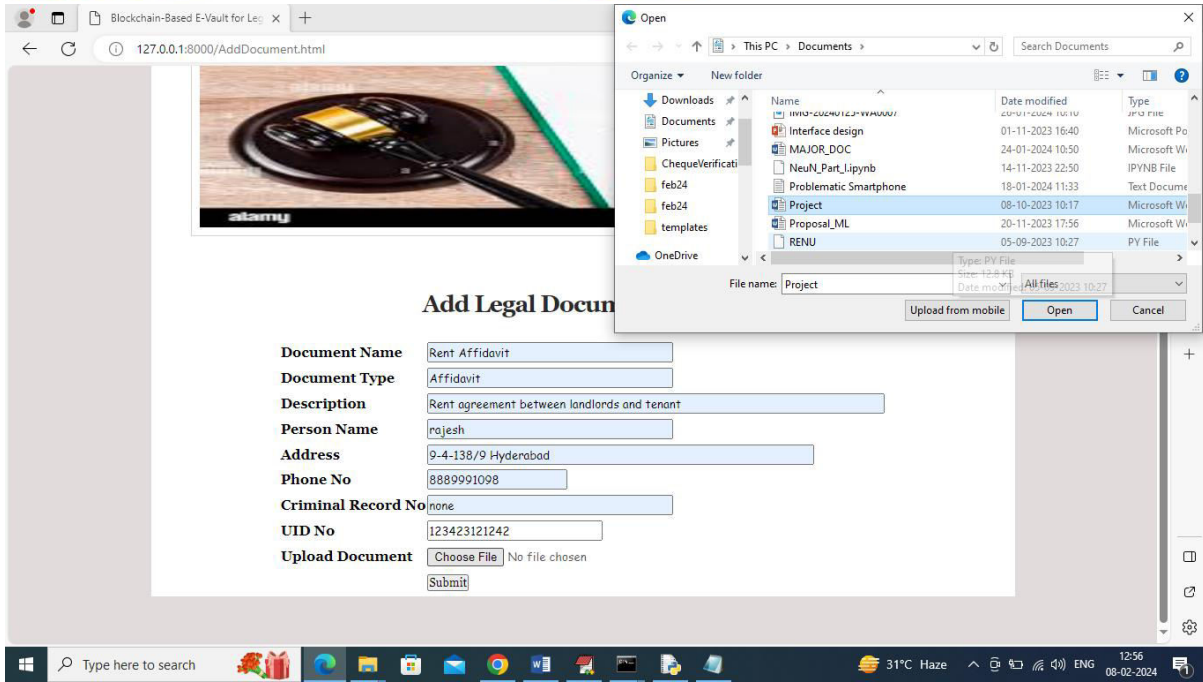
In above screen click on 'Admin Login Here' link to get below admin login page



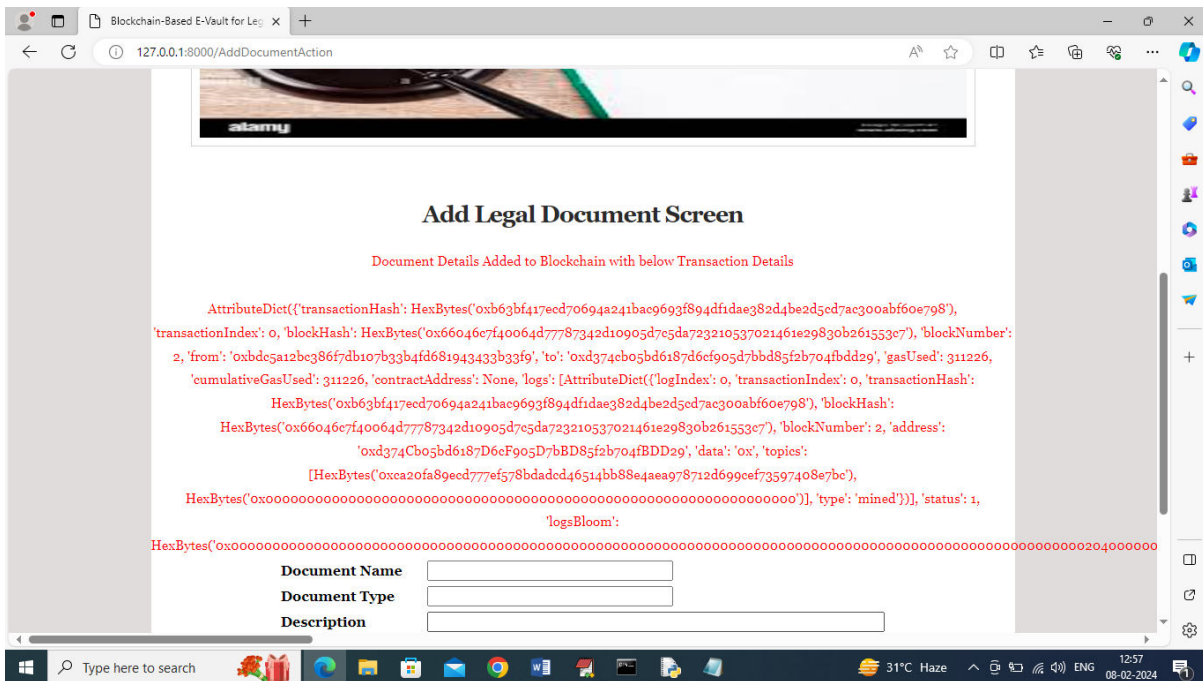
In above screen admin can login to system using username and password as 'admin' and then press button to get below page



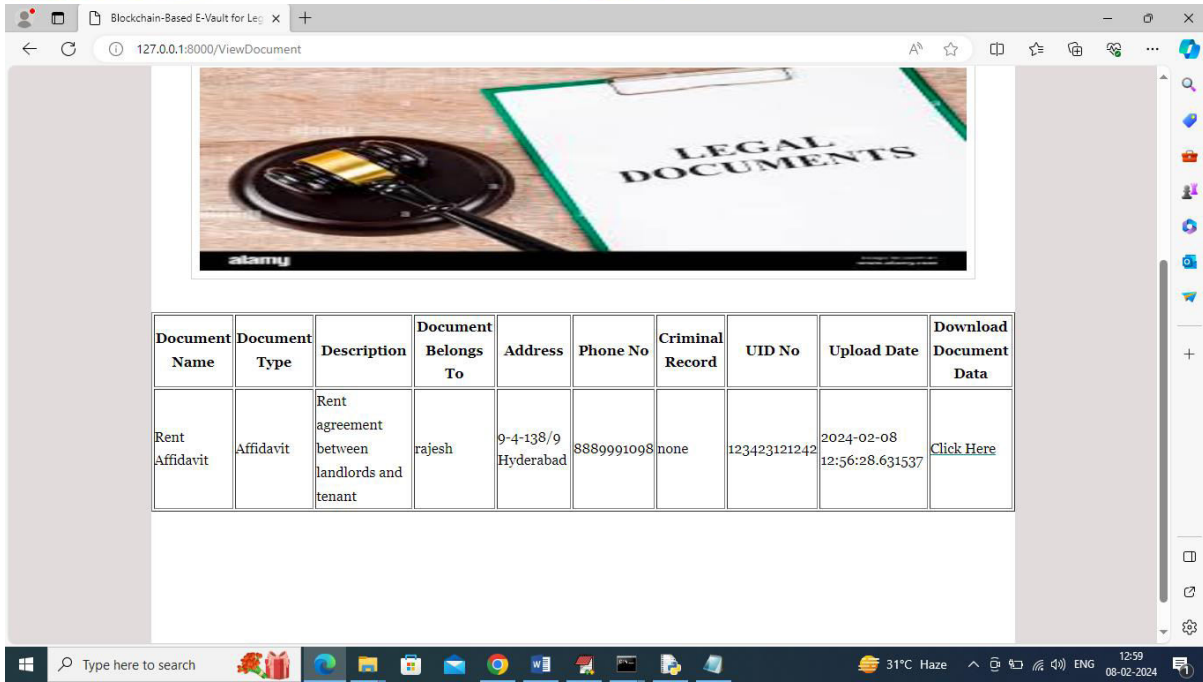
In above screen admin can click on 'Add Legal Document' link to add legal document details



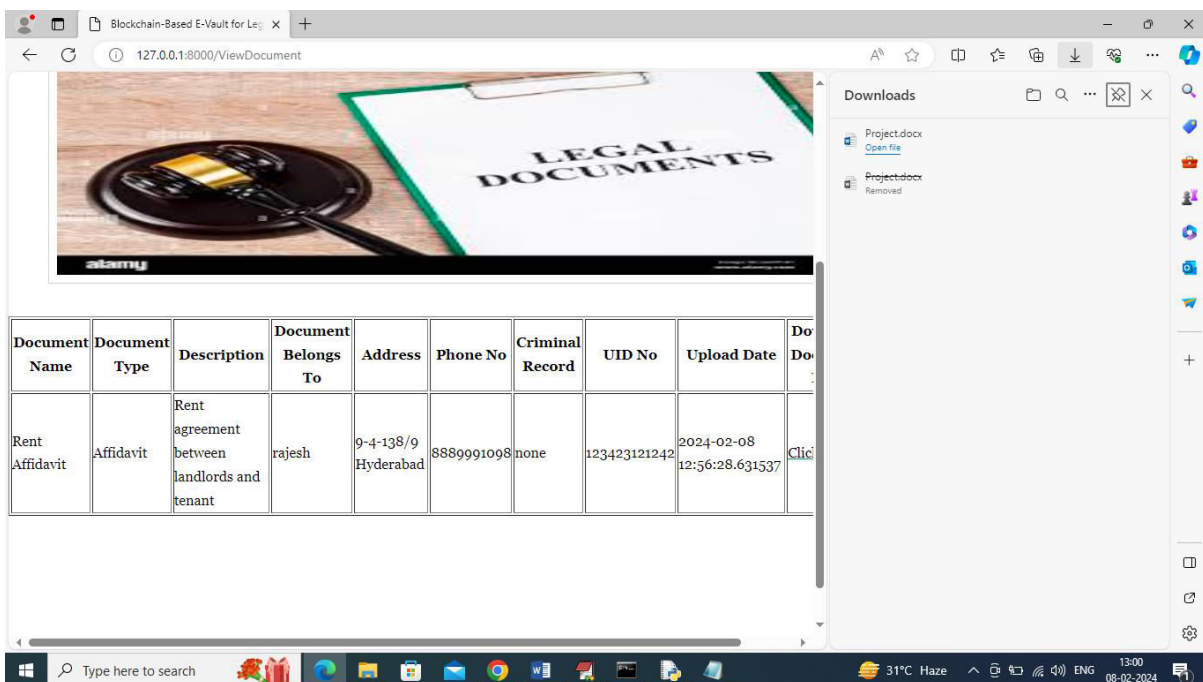
In above screen entering some type of Rent legal document details and then upload related document data and then click on 'Submit' button to save data in Blockchain and then will get below output



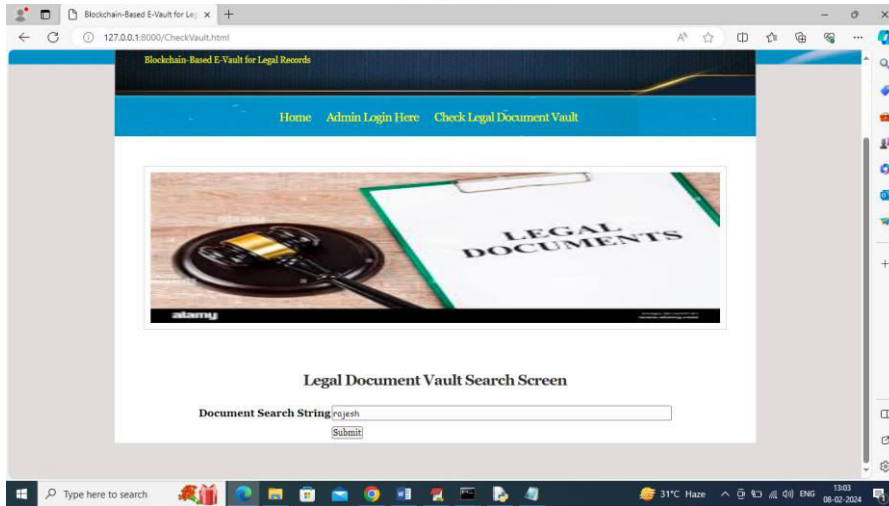
In above screen in red colour text can see all output returned from Blockchain after storage, normally will show transaction hash code but for you and your guide understanding we are displaying all details. In above output you can see Hash code and Block number as the core output. Now admin click on 'View Document' link to view all stored documents details in Blockchain



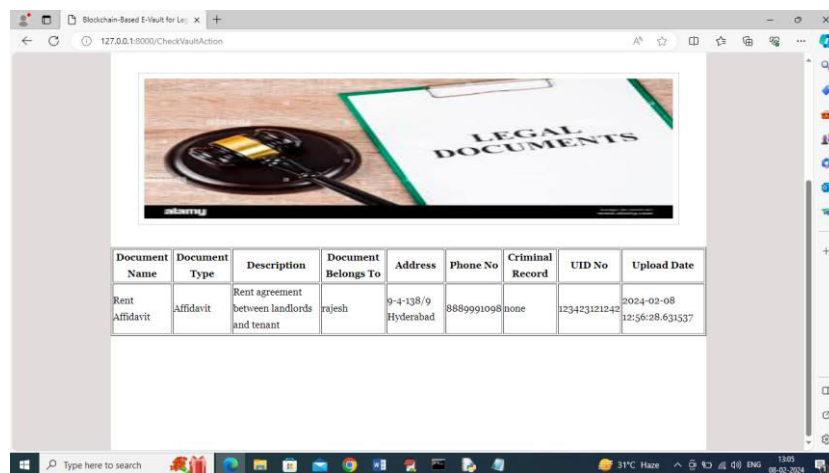
In above screen admin can view list of legal documents available in Blockchain and can click on 'Click Here' link to download associated document data file.



In above screen in right side panel can see downloading of associated data. Now logout and then normal users can view all available legal documents but we are granting to them to document details but cannot download associated data. In below screen normal users can search for legal documents



In above screen user can enter any string like person name about to search, document name or UID number or any other text. Entered input will be matched with available documents in Blockchain and get below result. In above screen I entered query as ‘rajesh’ means I want search any legal document exists on rajesh name



In above screen for given query user can see one search found and users are granting to view above details and cannot download associated legal document file.

So by using above screens you can manage all your documents in Blockchain.

6.0 CONCLUSION :

In conclusion, the proposed blockchain-based e-vault system represents a transformative solution for modernizing legal record management practices. By leveraging the inherent features of blockchain technology, including security, transparency, efficiency, and data privacy, the system offers significant advantages over existing record management practices. Through the use of cryptographic encryption, decentralized storage, and smart contracts, the e-vault system ensures the security and integrity of legal records, reducing the risk of unauthorized access, tampering, or forgery. The immutability and transparency of records stored on the blockchain enhance trust and accountability among stakeholders, while automation streamlines administrative tasks and accelerates the processing of legal records.



Furthermore, the e-vault system prioritizes data privacy and control, with advanced encryption and authentication mechanisms protecting sensitive information and ensuring compliance with data protection regulations. Stakeholders benefit from increased accessibility, convenience, and cost reduction, as they can securely access, manage, and verify their legal records from anywhere at any time. Overall, the proposed blockchain-based e-vault system offers a robust, efficient, and secure platform for managing legal records, enhancing trust and confidence in the legal system. By embracing innovative technologies and modernizing record management practices, stakeholders in the legal domain can optimize their operations, improve productivity, and deliver better outcomes for their clients and organizations.

REFERENCES :

1. Casey, A. J., Thompson, J., & Veldhuis, T. (2018). Blockchain technology: Balancing risks and opportunities. *Journal of Risk and Financial Management*, 11(4), 80.
2. O'Dwyer, R. E., Malone, D., & Gray, A. (2020). Blockchain in legal education: How will it revolutionise the industry? *International Journal of Law and Information Technology*, 28(3), 245-267.
3. De Filippi, P., & Wright, A. (2018). *Blockchain and the law: The rule of code*. Harvard University Press.
4. Bähring, A. (2019). Legal tech and blockchain: A new era for smart contracts and the legal industry. *Business Law Review*, 40(1), 3-37.
5. Hardjono, T., & Smith, D. W. (2018). *Introduction to decentralized identity*. MIT Connection Science & Engineering.
6. Yu, H., Kim, J., Kim, H., & Lee, D. H. (2019). A blockchain-based decentralized data storage and access framework for electronic health records. In *2019 International Conference on Information and Communication Technology Convergence (ICTC)* (pp. 1485-1487). IEEE.
7. Mödersheim, S., Samelin, K., & Strufe, T. (2017). Blockchain and the GDPR: Solutions for a responsible European data economy. In *European Data Protection: Coming of Age* (pp. 151-170). Springer.
8. Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. (2016). Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *2016 IEEE Symposium on Security and Privacy (SP)* (pp. 839-858). IEEE.
9. Werbach, K. (2018). *The blockchain and the new architecture of trust*. MIT Press.
10. Auerbach, D., Bauer, B., & Müller-Bloch, C. (2019). Blockchain and GDPR: How to reconcile privacy and distributed ledgers. In *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)* (pp. 87-96). IEEE.