Name of Student: Pushkar Sane		
Roll Number: 45	**Lab Assignment Number: 10**	
Title of Lab Assignment: Create Dapps in Ethereum.		
DOP: 20-10-2024	**DOS: 28-10-2024**	
CO Mapped:	**PO Mapped:**	**Signature:**

Practical No. 10

Aim: Create Dapps in Ethereum.

Theory:

1. **Introduction**
    a. **Purpose**

    VacChain aims to provide a blockchain-based solution for managing and verifying vaccine certifications. The project offers peer-to-peer verification, secure data storage, and decentralized handling of vaccine-related records.

    b. **Scope**
        i. Blockchain technology ensures tamper-proof, transparent vaccine records.
        ii. Supports user registration, login, and certificate uploads.
        iii. Offers dashboards for users and administrators.
        iv. Peer nodes communicate for record synchronization.
        v. Microservices handle various operations independently for scalability.
    c. **Definitions, Acronyms, and Abbreviations**
        i. Blockchain: Distributed ledger for secure and decentralized data management.
        ii. EJS: Embedded JavaScript templating for generating HTML pages.
        iii. Node: A peer in the blockchain network responsible for validating and syncing data.
        iv. Microservice: A small, independent service managing specific tasks.
2. **Overall Description**
    a. **Product Perspective**

    VacChain integrates blockchain with traditional web services for managing vaccine certificates. The system consists of:

    - Client-side: User interfaces for interacting with the platform.
    - Server-side: Blockchain nodes and microservices that ensure backend data integrity and synchronization

    b. **Product Features**
        i. User Management: Registration, login, and profile management.
        ii. Certificate Management: Upload, store, and validate vaccine certificates.
        iii. Dashboard: Displays vaccine-related statistics and user details.

iv. Peer Communication: Nodes exchange information to maintain a consistent blockchain ledger.

c. User Classes and Characteristics

i. End Users: Individuals uploading and verifying vaccine certificates.
ii. Administrators: Manage platform operations, users, and blockchain nodes.
iii. Peers: Other nodes in the network handling data synchronization.

3. Functional Requirements

a. User Registration and Login

i. FR1: Users should be able to register using email and password.
ii. FR2: Users should be able to log in and access their dashboard.

b. Certificate Management

i. FR3: Users can upload vaccine certificates in PDF format.
ii. FR4: The system should validate uploaded certificates against a predefined schema.

c. Dashboard

i. FR5: Users and admins should have separate dashboards displaying relevant statistics.
ii. FR6: Admins can view and manage uploaded certificates.

d. Blockchain Node Communication

i. FR7: Peers should exchange data to ensure blockchain consistency.
ii. FR8: Microservices handle individual components like authentication and peer updates.

4. Non-Functional Requirements

a. Performance Requirements

i. The system should handle up to 100 concurrent users without performance degradation.
ii. Data synchronization between peers should complete within 2 seconds.

b. Security Requirements

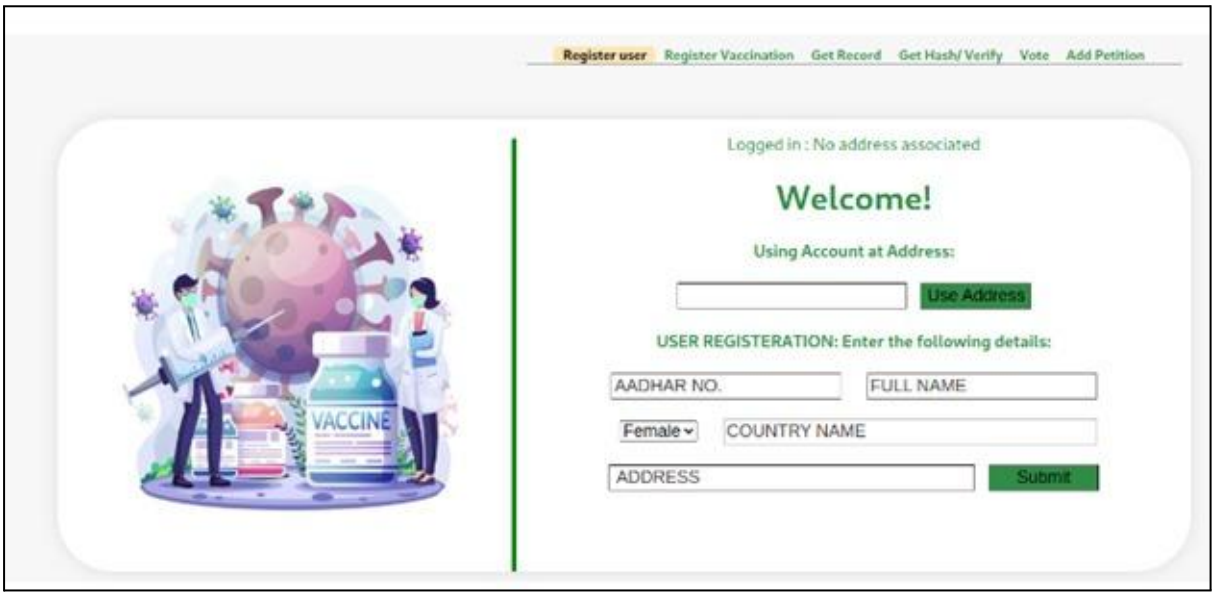
i. All sensitive data, such as passwords, should be encrypted.
ii. Blockchain nodes should only communicate over secure channels.

c. Usability Requirements

i. The UI should be responsive and work on desktops and mobile devices.
ii. Certificate uploads should provide feedback within 5 seconds.

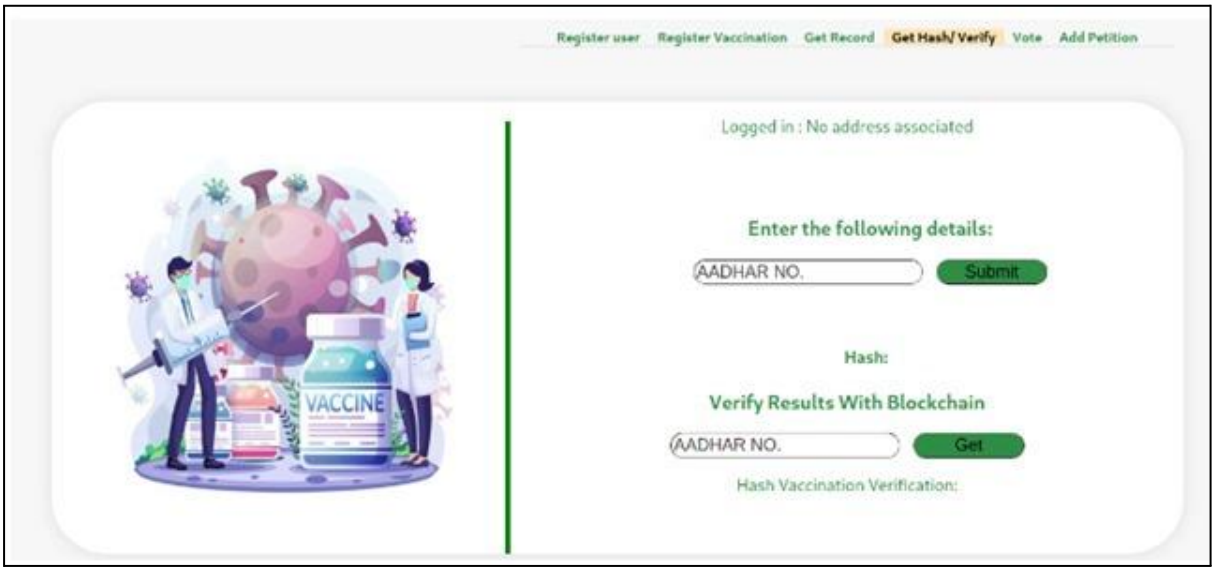
5. **System Architecture**
    - Frontend: EJS-based templates for login, registration, and dashboards.
    - Backend: Node.js-based microservices handling various functionalities.
    - Blockchain: Custom blockchain logic implemented for certificate storage
6. **Constraints**
    - The blockchain ledger cannot be altered once data is stored.
    - Each node must remain online to maintain network integrity
7. **Assumptions and Dependencies**
    - All users must have internet access to interact with the system.
    - The blockchain nodes need to be online for smooth operation.
8. **Appendix**
    - Technologies Used: Node.js, EJS, Blockchain, Microservices.
    - Future Enhancements: Support for mobile app integration, advanced analytics on vaccine data.

Output:




Register user
Register Vaccination
Get Record
Get Hash/ Verify
Vote
Add Petition
Logged in : No address associated
Register Vaccination
Enter Vaccination Details:
AADHAR NO.
VACCINATION NAME
BATCH ID
LOCATION
mm / dd / yyyy
Submit
VACCINE


Register user
Register Vaccination
Get Record
Get Hash/ Verify
Vote
Add Petition
Logged in : No address associated
Only Vaccination Authority/User
AADHAR NO.
Get
Click to view details!
VACCINE



Conclusion:

VacChain is a secure, blockchain-based system for managing vaccine certificates, providing tamper-proof storage and peer-to-peer verification. Users can register, upload certificates, and access dashboards, while administrators manage platform operations. The platform uses microservices for scalability and real-time data synchronization across blockchain nodes. Future enhancements include mobile support and analytics for expanded functionality.