

BCertiFY

KG1 : Blockchain Based Certificate Specification



GROUP MEMMBERS

Kakumanu Vamsee Krishna
22111065

Arabinda Karmakar
22111011

Pratik Mahipal Patil
22111047

Bharat Kumar Koduru
22111075

Date - 19/04/2023



Tech Stack



Objectives Implemented



**Distribution of
Information**

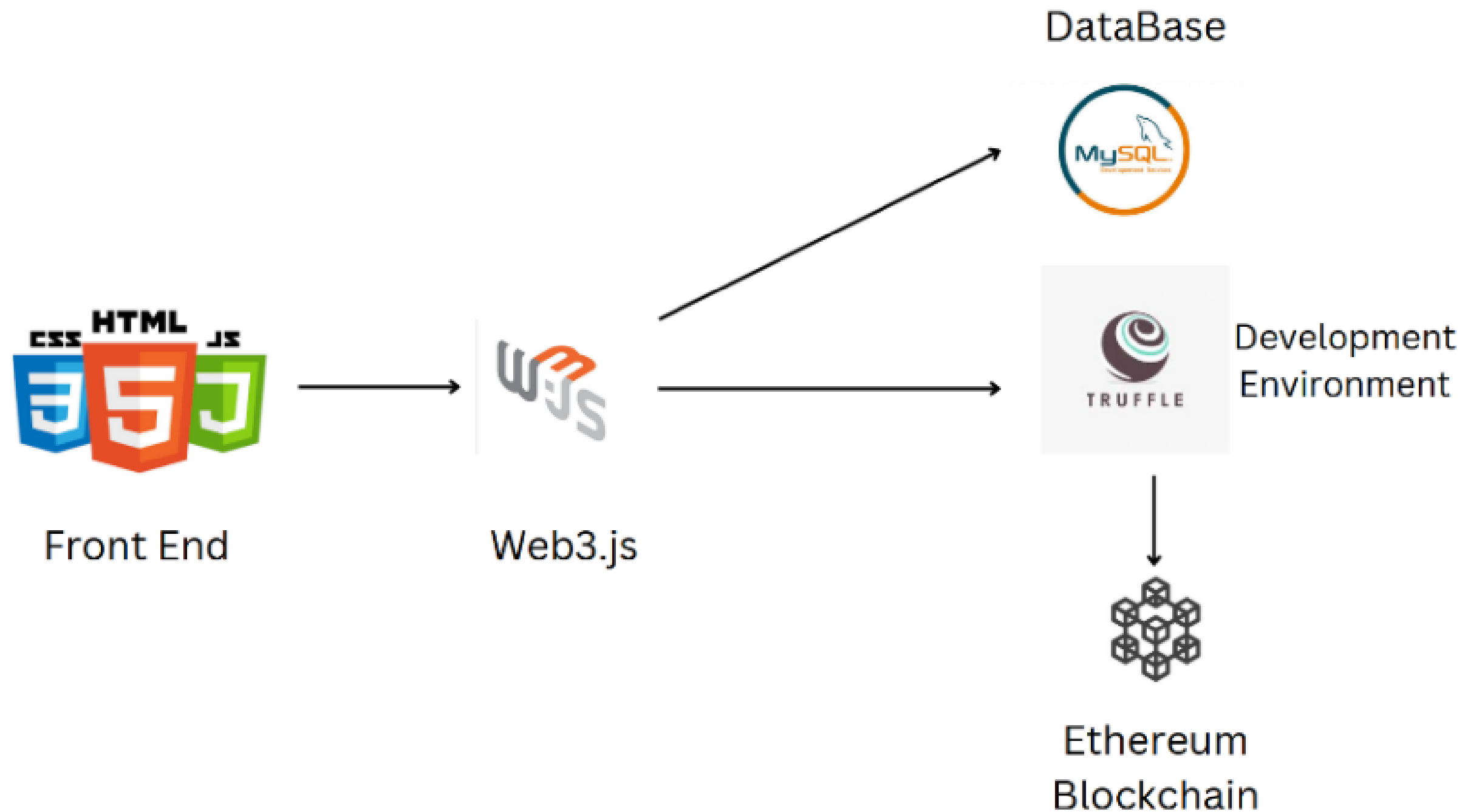


Prototype Workflow



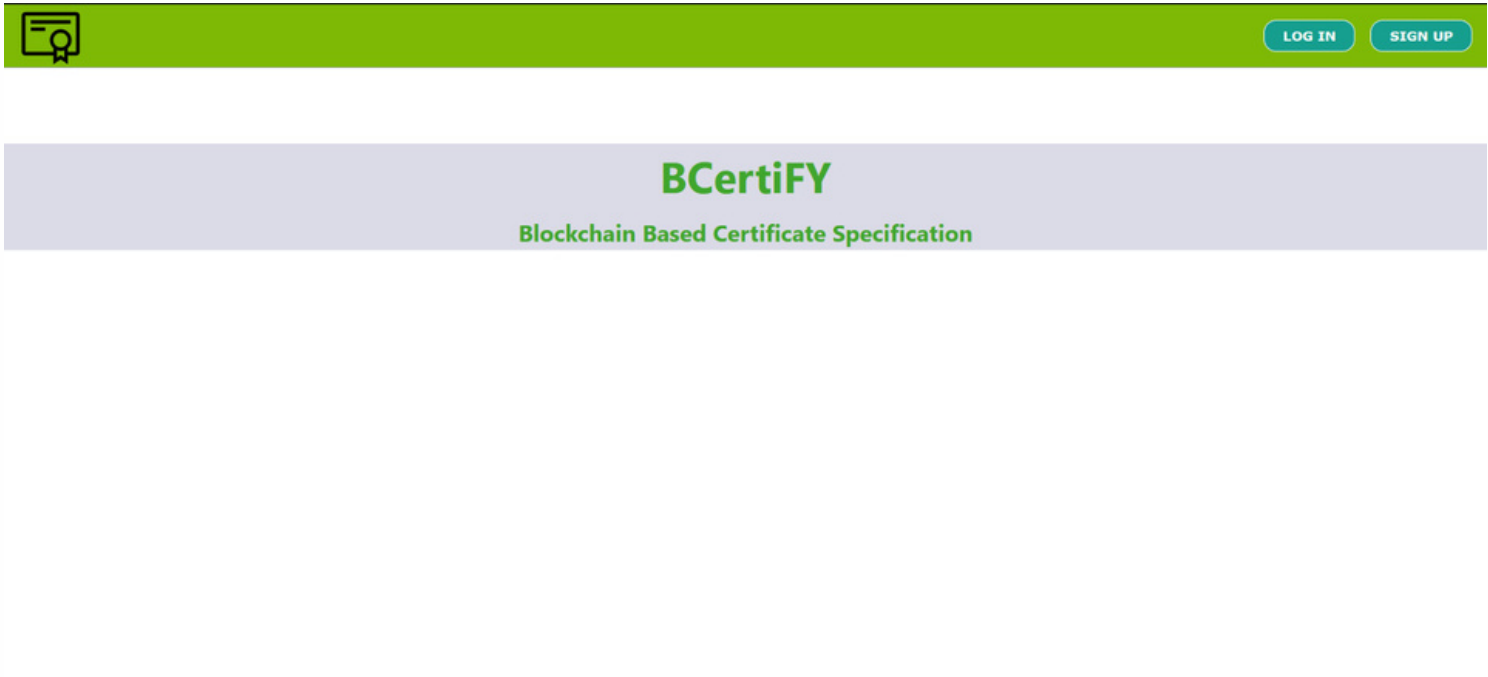
Contribution of Group Members

Tech Stack

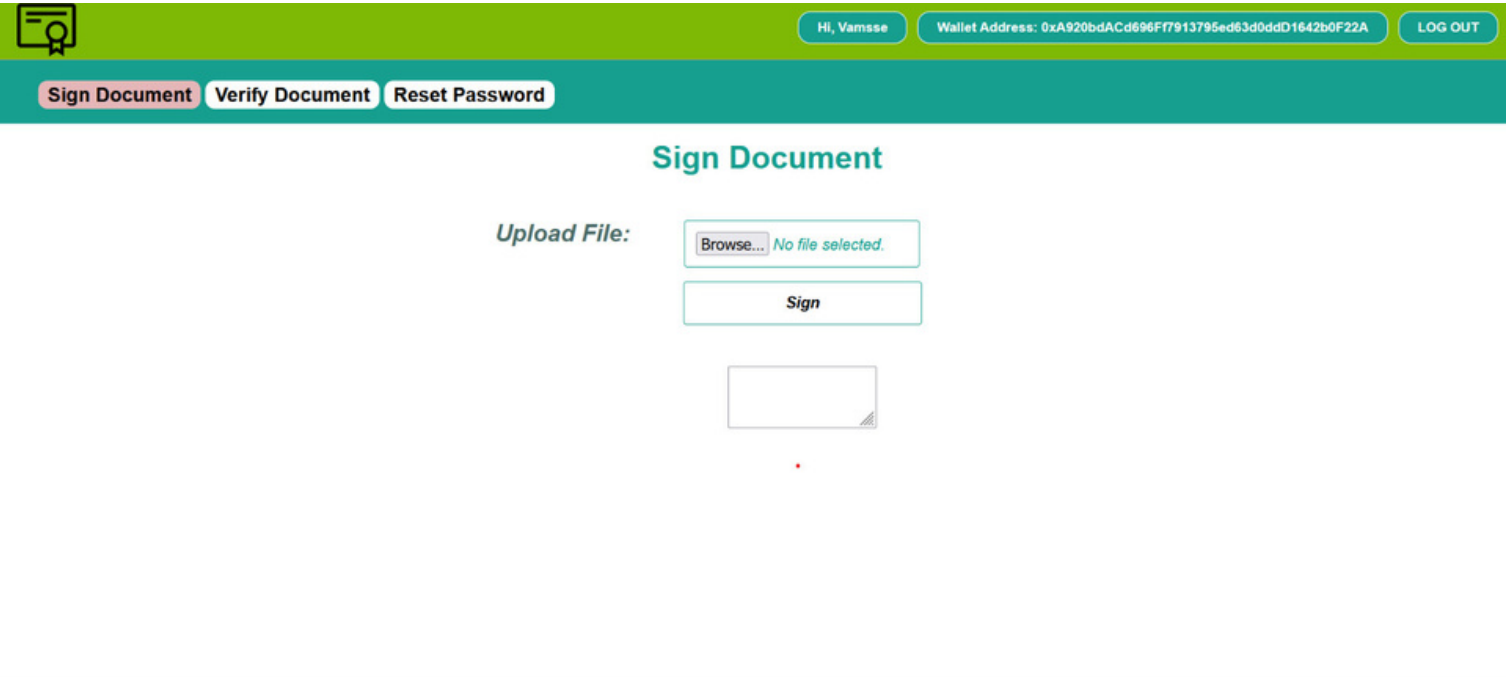


Front End

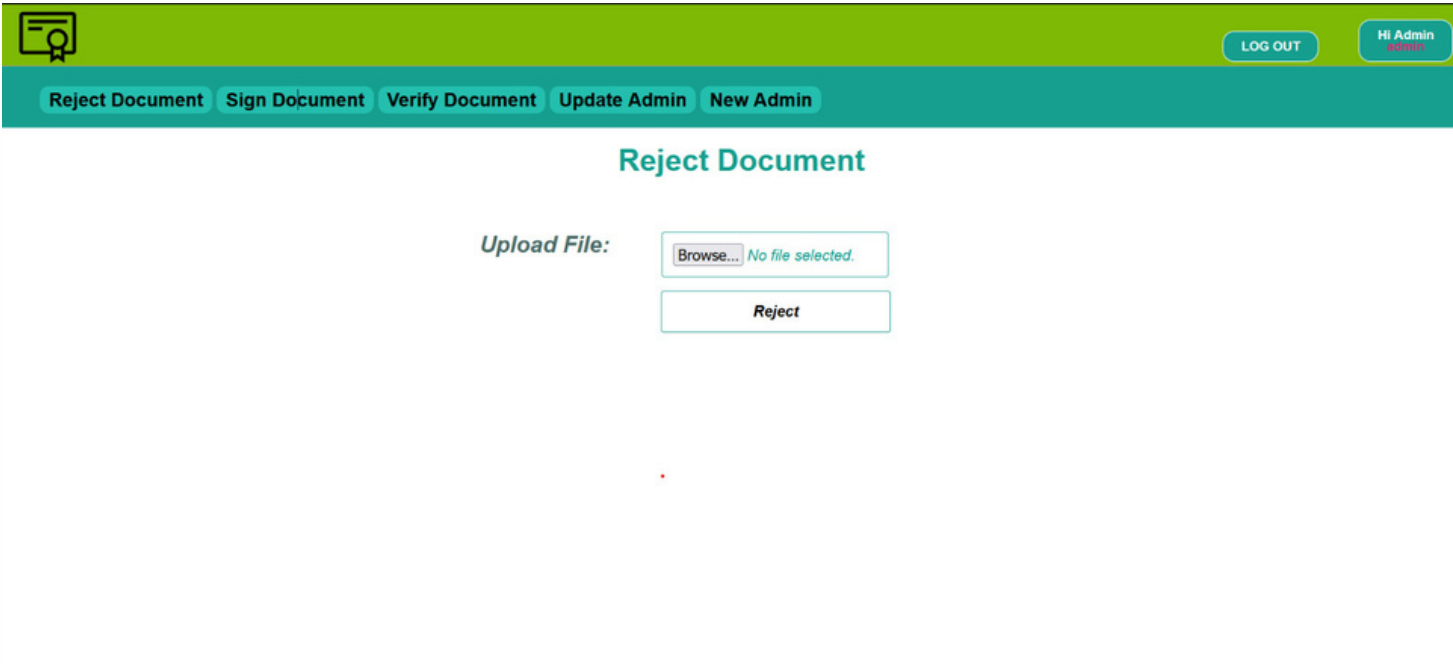
Index Page:



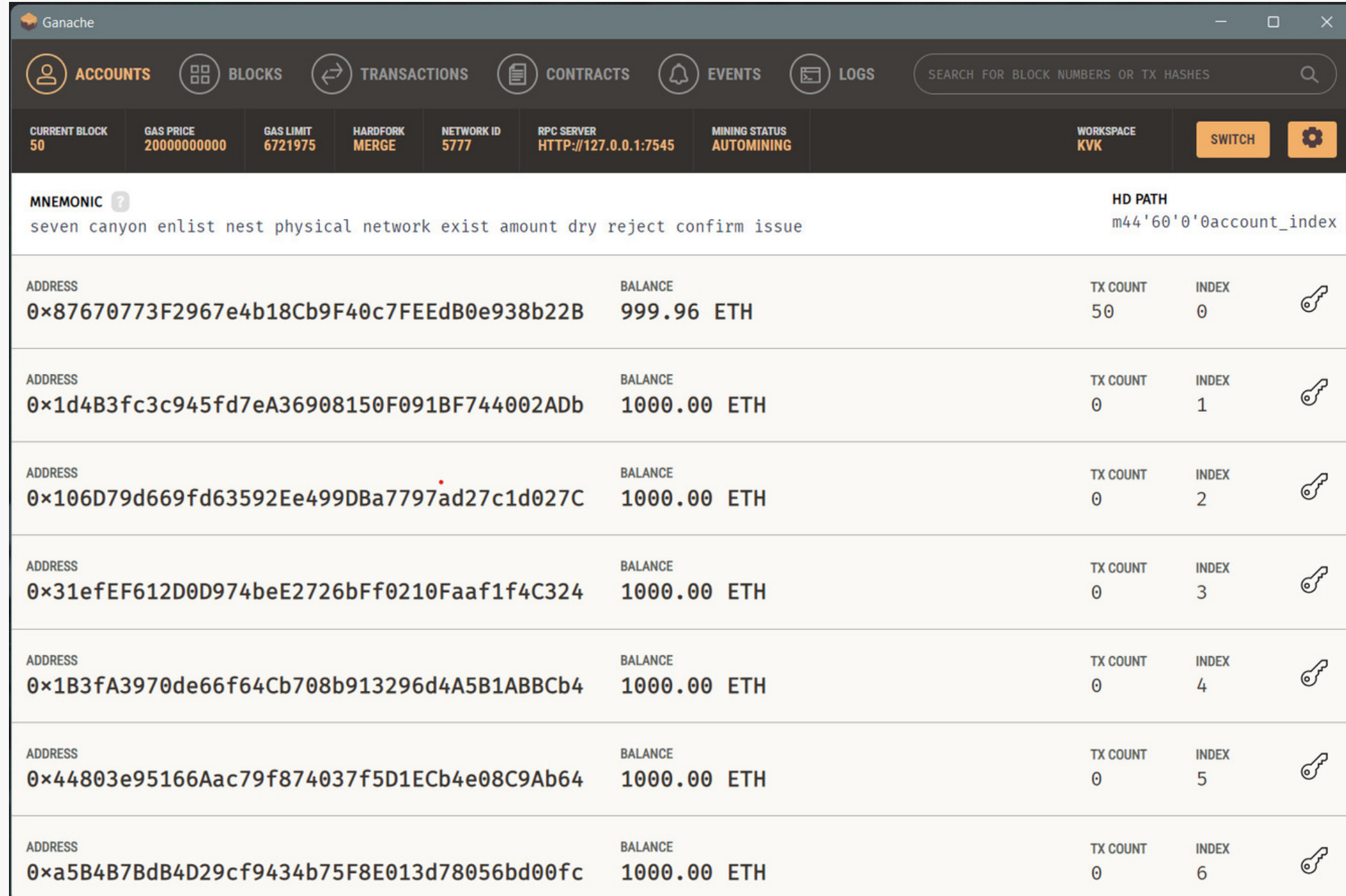
Userpanel:



Admin Panel



Setup Ganache



1.Set up the private blockchain using Ganache

2. Configured by <http://127.0.0.1:7545>

3. Deployed the smart contract.










Smart Contract

```
struct Certificate {  
    uint256 id;  
    bytes32 hash;  
    address issuer;  
    bool isSigned;  
}
```

Functions

1. signCertificate ()
2. rejectCertificate ()
3. verifyCertificate()

What Has been Implemented?

Project Requirements	New Features added
1. No Need to Carry Physical Certificate (use public key of user to verify) 	1.Informing the user about the sign/verification /Reject Status of Certificate *** 
2.Anyone can create an account on app 	2.Admin panel 
3.Immutability of the certificates using Blockchain. 	3.Forgot Password/Reset Password of user 
4.Anyone can verify the certificate 	
5.Remove/Reject Fake Certificates 	
6.A User can get a certificate signed to anyone to their public key (User) 	

***Web3.js Connection with front-end and back-end has been implemented but not working.

Strategies and Techniques

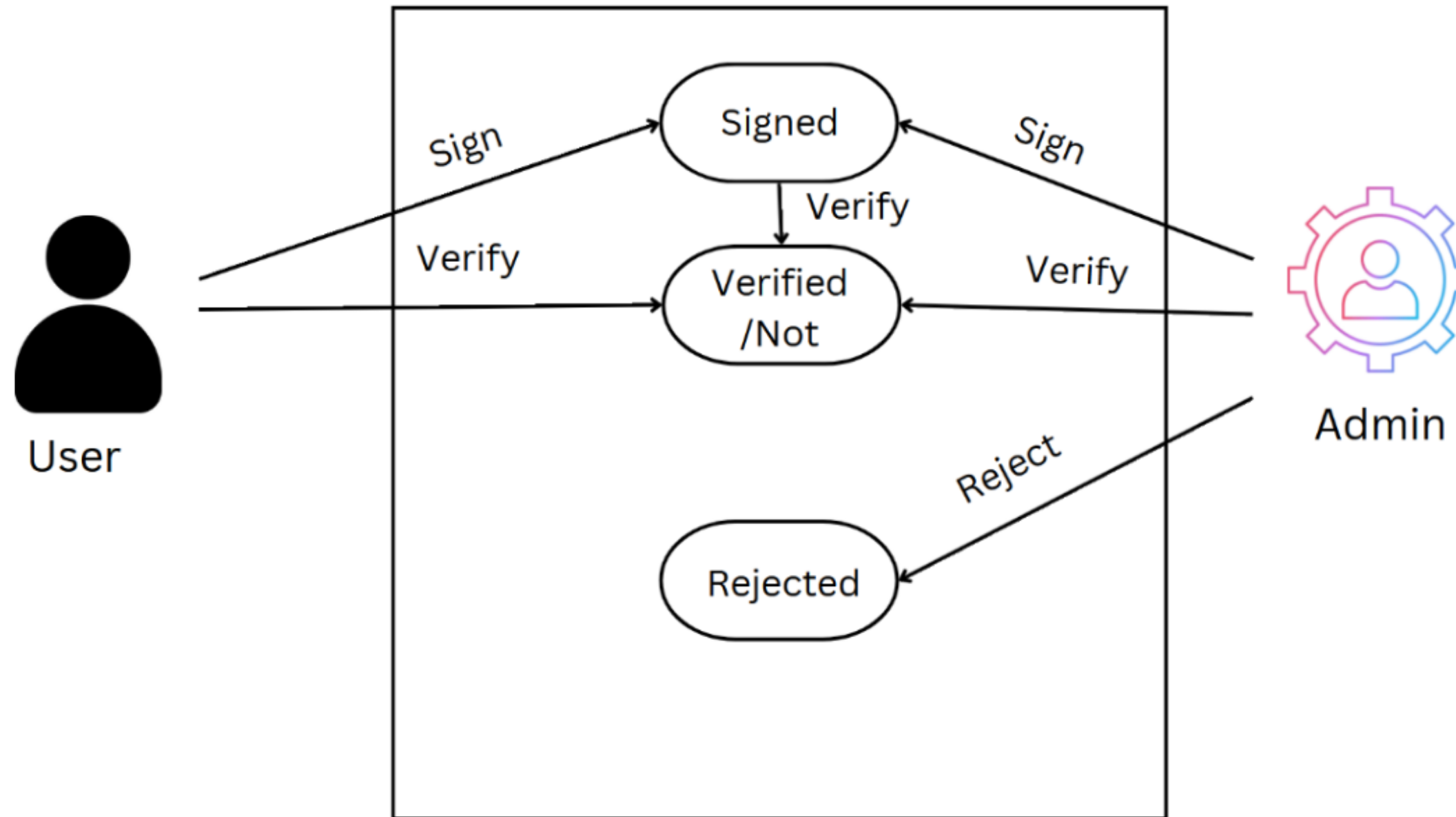
On-Chain

- 1.Certificate Id
- 2.MD5 Hash of the Certificate
- 3.isSigned Boolean
- 4.Issuers Address
- 5.CertificatesIssued - Mapping
- 6.SignedCertificates - Mapping
- 7.RejectedCertificates - Mapping

Off-Chain

- 1.User and Admin Login Credentials
- 2.User Wallet Address

Work Flow



Contributions

Kakumanu Vamsee Krishna

- 1.Setting up the environment
- 2.Smart contract for the Dapp (Verify)
3. Integration testing.***

Arabinda Karmakar

1. Database connection to the Front-end.
- 2.Smart contract for the Dapp(Sign)
3. Integration testing.***

Pratik Mahipal Patil

- 1.Front-End of the DApp
2. Deployment of Smart Contract
- 3.Smart contract for the Dapp (Verify)

Bharat Kumar Koduru

- 1.Database Design
- 2.Front-End of the DApp
- 3.Smart contract for the Dapp (Reject)

Thank You

Any Questions?