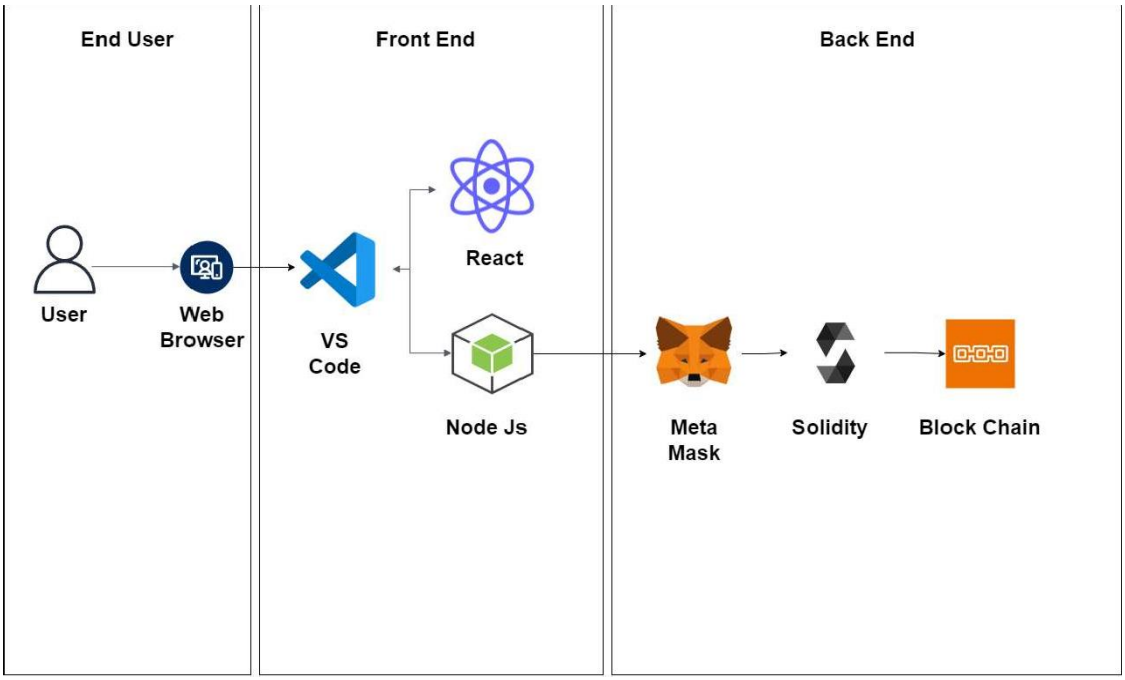
Project DesignPhase-I

Solution Architecture

TeamID	NM2023TMID04387
Project Name	Project - Ethereum Decentralised Identity Smart Contract
Maximum Marks	4Marks

Solution Architecture:



Solution architecture Description:

Front End:

User Interface (UI):

- The frontend development phase begins with creating a user-friendly interface for users to interact with the Ethereum DID smart contract. This includes web or mobile applications where users can manage their identities, keys, and transactions.

Wallet Integration:

- Integration with Ethereum wallets is essential to manage and sign transactions. Ethereum wallets allow users to interact with the

smart contract securely and sign messages for identity verification**.**

DID Management:

- Frontend interfaces should provide users with the ability to create, update, and revoke their DIDs. These interfaces may include forms for inputting identity data and options for managing keys and related information.

Interaction with Smart Contracts:

- Frontend development involves writing code that communicates with Ethereum smart contracts. This typically involves using web3 libraries to send transactions, query contract data, and perform various DID-related operations.

BACKEND:

Smart Contract Development:

- The core of Ethereum DID systems is the smart contract that manages DIDs, keys, and related data. This phase involves writing, testing, and deploying smart contracts to the Ethereum blockchain. Smart contracts handle the validation, creation, and management of DIDs.

Blockchain Integration:

- Backend development also includes integration with the Ethereum blockchain. This includes setting up infrastructure to interact with the Ethereum network and handling events emitted by the smart contract.

Identity Verification Services:

- Some Ethereum DID systems may incorporate identity verification services, which could be part of the backend. These services may include mechanisms for verifying the authenticity of identity claims.

Encryption and Security:

- Ensuring the security of user data and keys is a crucial aspect of backend development. Backend components may involve encryption, secure storage, and key management.

Event Handling:

- Smart contracts emit events for various actions, and the backend should have mechanisms to listen to and handle these events for notifying users of changes in their DIDs.