CODTECH INTERNSHIP - TASK 4

# Private Blockchain Implementation

Name: Vaahini D

Intern ID: CT04DG1604

## Overview

This document covers the setup and deployment of a private blockchain using Hyperledger Fabric. A sample decentralized application (DApp) is deployed on the blockchain.

## Setup Instructions

1. Install prerequisites: Docker, Docker-Compose, Node.js, Go, and Hyperledger Fabric binaries.

2. Generate crypto materials and genesis block using cryptogen and configtxgen.

3. Set up Docker containers for orderers, peers, and CLI using docker-compose.

4. Create a channel, join peers, and install chaincode (smart contract).

5. Instantiate chaincode and invoke/query it using CLI or a Node.js SDK.

## Sample Chaincode (Smart Contract)

// SPDX-License-Identifier: Apache-2.0  
'use strict';  
  
const { Contract } = require('fabric-contract-api');  
  
class SimpleContract extends Contract {  
 async initLedger(ctx) {  
 console.info('Ledger initialized');  
 }  
  
 async createAsset(ctx, assetId, value) {  
 const asset = {  
 value: value,  
 };  
 await ctx.stub.putState(assetId, Buffer.from(JSON.stringify(asset)));  
 return `Asset ${assetId} created`;  
 }  
  
 async readAsset(ctx, assetId) {  
 const assetJSON = await ctx.stub.getState(assetId);  
 if (!assetJSON || assetJSON.length === 0) {  
 throw new Error(`Asset ${assetId} does not exist`);  
 }  
 return assetJSON.toString();  
 }  
}  
  
module.exports = SimpleContract;

## Expected Output & Results

- Blockchain network successfully created using Hyperledger Fabric.

- Smart contract deployed and invoked.

- Assets created and retrieved using chaincode.

- Working DApp successfully connected with blockchain network and invoked chaincode functions.