# Decentralized Identity Management System (DIMS) - Intern Guidance Document

This document provides detailed step-by-step guidance for interns working on the Decentralized Identity Management System (DIMS) project using the MERN stack and blockchain technology. The project is divided into four roles: Blockchain Developer, Backend Developer, Frontend Developer, and Security Analyst. Each role has specific tasks, tools, commands, and expected outcomes over a 4-week timeline.

## Blockchain Developer

### Week 1

* - Install and configure Hardhat or Truffle for smart contract development.
* - Write Solidity smart contracts for user registration and role-based access control.
* - Test contracts locally using Ganache.

### Week 2

* - Deploy smart contracts to local blockchain.
* - Integrate MetaMask for wallet-based authentication.
* - Emit events for user actions (e.g., login, role assignment).

### Week 3

* - Refactor contracts for modularity and security.
* - Document contract functions and expected inputs/outputs.

### Week 4

* - Assist in integration with frontend and backend.
* - Conduct final testing and debugging of smart contracts.

## Backend Developer

### Week 1

* - Set up Node.js and Express.js server.
* - Design RESTful API endpoints for user registration and login.
* - Connect to MongoDB using Mongoose.

### Week 2

* - Integrate Web3.js or Ethers.js to interact with smart contracts.
* - Implement JWT-based session management.
* - Create middleware for role-based access control.

### Week 3

* - Secure API endpoints using Helmet.js and bcrypt.
* - Log user actions and store metadata in MongoDB.

### Week 4

* - Perform integration testing with frontend and blockchain.
* - Optimize performance and document API usage.

## Frontend Developer

### Week 1

* - Initialize React.js project and set up routing.
* - Design UI components for registration and login.
* - Install MetaMask and configure Web3.js integration.

### Week 2

* - Connect frontend to backend APIs and smart contracts.
* - Implement role-based dashboards (Admin/User/Guest).

### Week 3

* - Display audit logs and access control feedback.
* - Ensure responsive design and user experience.

### Week 4

* - Conduct usability testing and fix UI bugs.
* - Prepare demo and walkthrough of the application.

## Security Analyst

### Week 1

* - Define IAM policies and role definitions.
* - Create access control matrix for the system.

### Week 2

* - Review smart contracts for security vulnerabilities.
* - Audit backend API for secure data handling.

### Week 3

* - Monitor access logs and flag anomalies.
* - Implement basic security alerts and logging.

### Week 4

* - Prepare security report and recommendations.
* - Assist in final security testing and documentation.

# ✅ Weekly Progress Checklist

## Blockchain Developer

### Week 1

☐ Install and configure Hardhat or Ganache

☐ Design initial smart contract structure

☐ Test basic contract deployment

### Week 2

☐ Implement user registration in smart contract

☐ Add role-based access control logic

☐ Deploy updated contract locally

### Week 3

☐ Integrate smart contract with backend via Web3.js

☐ Test MetaMask login functionality

☐ Log events on-chain

### Week 4

☐ Finalize smart contract code

☐ Conduct security audit of contract

☐ Document smart contract functions

## Backend Developer

### Week 1

☐ Set up Express.js server

☐ Connect to MongoDB using Mongoose

☐ Define user schema

### Week 2

☐ Create RESTful APIs for user registration/login

☐ Implement JWT-based session management

☐ Secure endpoints with middleware

### Week 3

☐ Integrate Web3.js to interact with smart contract

☐ Store audit logs in MongoDB

☐ Test role-based access control

### Week 4

☐ Optimize API performance

☐ Write backend documentation

☐ Prepare deployment scripts

## Frontend Developer

### Week 1

☐ Initialize React project

☐ Design UI wireframes

☐ Set up routing and layout

### Week 2

☐ Build registration and login components

☐ Integrate MetaMask login

☐ Display user roles and permissions

### Week 3

☐ Connect frontend to backend APIs

☐ Render audit logs and access control messages

☐ Test UI responsiveness

### Week 4

☐ Polish UI/UX design

☐ Fix bugs and finalize components

☐ Write frontend documentation

## Security Analyst

### Week 1

☐ Define IAM roles and permissions

☐ Draft access control matrix

☐ Review smart contract design

### Week 2

☐ Implement RBAC logic in backend

☐ Set up logging and monitoring

☐ Conduct basic vulnerability scan

### Week 3

☐ Audit smart contract and API security

☐ Analyze access logs for anomalies

☐ Prepare security checklist

### Week 4

☐ Finalize IAM policies

☐ Write security report

☐ Present findings to team