IBM Innovation Center for Education

PHEMESOFT
Software That Matters
TRUSTED DELIVERY PARTNER for IBM ICE

RABBI ZIDNI 'ILMA
YENEPOYA
(DEEMED TO BE UNIVERSITY)

YENEPOYA INSTITUTE OF ARTS, SCIENCE, COMMERCE AND MANAGEMENT

A CONSTITUENT UNIT OF YENEPOYA (DEEMED TO BE UNIVERSITY) BALMATTA, MANGALORE

BLOCKCHAIN FOR SECURE DIGITAL IDENTITY MANAGEMENT

PROJECT SYNOPSIS

BACHELOR OF COMPUTER APPLICATION

Cyber Forensics, Data analytics & Cyber Security

Submitted by:

Muneeba Marjan — 22BCACDC51

Guided by:

Ms. Prathiksha

TABLE OF CONTENTS

1. INTRODUCTION

Digital identity management faces critical challenges in today's interconnected world, with centralized systems proving vulnerable to data breaches (61% of 2023 cyber incidents according to Verizon DBIR). This project addresses these vulnerabilities by developing a blockchain-based decentralized identity management system (DIDMS) on the Ethereum platform, leveraging smart contracts to create tamper-proof digital identities.

Our solution implements a three-tier architecture:

1. **Blockchain Layer**: Ethereum testnet with Solidity smart contracts
2. **Application Layer**: Hardhat environment for deployment
3. **Interface Layer**: React frontend with MetaMask integration

Key technical features include:

- Self-sovereign identity creation using cryptographic keys
- Role-based access control (RBAC) for administrators
- Selective disclosure mechanisms
- Immutable audit trails

The system specifically targets:

- Elimination of single points of failure
- Reduction of identity fraud through cryptographic verification
- Improved user privacy controls

Aligned with W3C DID standards, this implementation demonstrates how blockchain technology can revolutionize identity management while complying with GDPR and CCPA regulations. The project combines Ethereum's smart contract capabilities with practical web3 development to create a working prototype for secure, decentralized identity verification.

2. METHODOLOGY/PLANNING OF WORK

Our development follows a structured 7-phase agile methodology:

*Phase 1: Requirement Analysis (Week 1-2)

- Conduct comparative analysis of existing solutions (uPort, Sovrin)
- Identify core functional requirements
- Define smart contract architecture
- Create UML diagrams for system workflow

*Phase 2: Smart Contract Development (Week 3-6)

- Write Solidity contracts (IdentityManagement.sol)
- Implement ERC-725 standard for identity management
- Develop verification/revocation mechanisms
- Unit testing with Hardhat (100% coverage target)

*Phase 3: Frontend Development (Week 7-9)

- Build responsive UI with Javascript and HTML
- Implement ethers.js for blockchain interaction
- Develop MetaMask integration flow
- Create admin dashboard with role-based views

*Phase 4: Testing & Deployment (Week 10-12)

- Comprehensive security audits (MythX)
- Gas optimization analysis
- Deployment to Goerli testnet

3. FACILITIES REQUIRED

Development Environment*:*

Hardware:

- Workstation (8GB RAM, SSD)
- Ethereum node (Infura/Alchemy)

Software Stack:

- Toolchain: Hardhat + Node.js v18
- Smart Contracts: Solidity 0.8.20+
- Testing: Chai+Mocha (600+ test cases)
- CI/CD: GitHub Actions

Security Infrastructure:

- Static Analysis: Slither, Mythril
- Dynamic Analysis: Hardhat Network
- Monitoring: Tenderly for tx tracing

Collaboration Tools:

- Version Control: GitHub (private repo)
- Documentation: Docusaurus
- Project Management: Jira

4. REFERENCES

Core Technical References:

1. Ethereum Yellow Paper (Wood, 2022)
2. ERC-725 Standard (Fabian Vogelsteller)
3. W3C DID Specification v1.0

Implementation Guides:

- Hardhat Developer Documentation (2023)
- MetaMask API Reference
- OpenZeppelin Contracts v4.9

Academic Sources:

- "Decentralized Identity" (IEEE Blockchain, 2022)
- "SSI Architectures" (Springer, 2021)