Blockchain-Driven Personal Data Vault
Full Implementation Guide

1. Introduction

This guide provides a comprehensive step-by-step approach to building a Blockchain-Driven Personal Data Vault with AI-Driven Access Control. The project aims to enhance privacy and security using blockchain technology for decentralized storage and AI for intelligent access control.

2. Setting Up Blockchain & Smart Contracts

We use Ethereum and Solidity to create a smart contract that allows users to securely store and share encrypted data.

Steps:
1. Install Node.js and Truffle for smart contract development.
2. Create a Solidity contract (`DataVault.sol`) that includes functions to:
   - Store encrypted data.
   - Grant and revoke access.
   - Retrieve stored data securely.
3. Deploy the contract on the Sepolia Testnet using Infura.

3. Developing the Backend API

A Node.js backend is developed using Express and Web3.js to allow interaction with the blockchain. It provides endpoints for:
- Storing encrypted data.
- Granting access to other users.
- Retrieving stored data securely.

The backend connects to the Ethereum blockchain and ensures smooth transaction handling using MetaMask authentication.

4. Implementing AI-Driven Access Control

AI is integrated to improve access security by analyzing user behavior, contextual factors (time, device trust score), and previous access patterns.

Steps:
1. Train a RandomForestClassifier model using sample user behavior data.
2. Deploy the trained AI model using Flask.
3. Integrate AI predictions into the backend API to determine access control dynamically.

5. Developing the React Frontend

A React web application is created to allow users to:

- Connect their MetaMask wallet.
- Store and retrieve encrypted data.
- Grant and revoke access using AI verification.

Web3.js is used to connect the React UI to the Ethereum smart contract.

6. Deploying the System

The final step is deploying both the smart contract and the React web app:

1. Deploy the smart contract on Sepolia using Truffle.
2. Deploy the React web app using Netlify or Vercel.
3. Test the complete system to ensure smooth functionality.

7. Conclusion

By integrating blockchain with AI, this system enhances privacy and security for personal data management. The decentralized architecture ensures data integrity, while AI dynamically adapts access control policies for improved security.