CredChain - Web3 Employment & Academic Credentials Verification Platform

Submission Date - 04 May 2025

By- 1. Pappu Kumar
2. Harsh Parihar

Agenda

A

Introduction

A overview of project and Motivate our problem statement.

B

System Design & Tech Stack

One Slide Introduction to CredChain and Discuss the Tech Stack used.

C

Modules Implemented & Details

Introduction to module implemented and explain the implementation details.

D

Demos, code walkthrough and Future Plans

Show the Demos and code walkthrough & Discuss future plans.

Problem with current credential verification system

A **Credential Fraud & No Portability**

B **Inefficient Verification Processes**

C **Data Silos and Privacy Concerns**

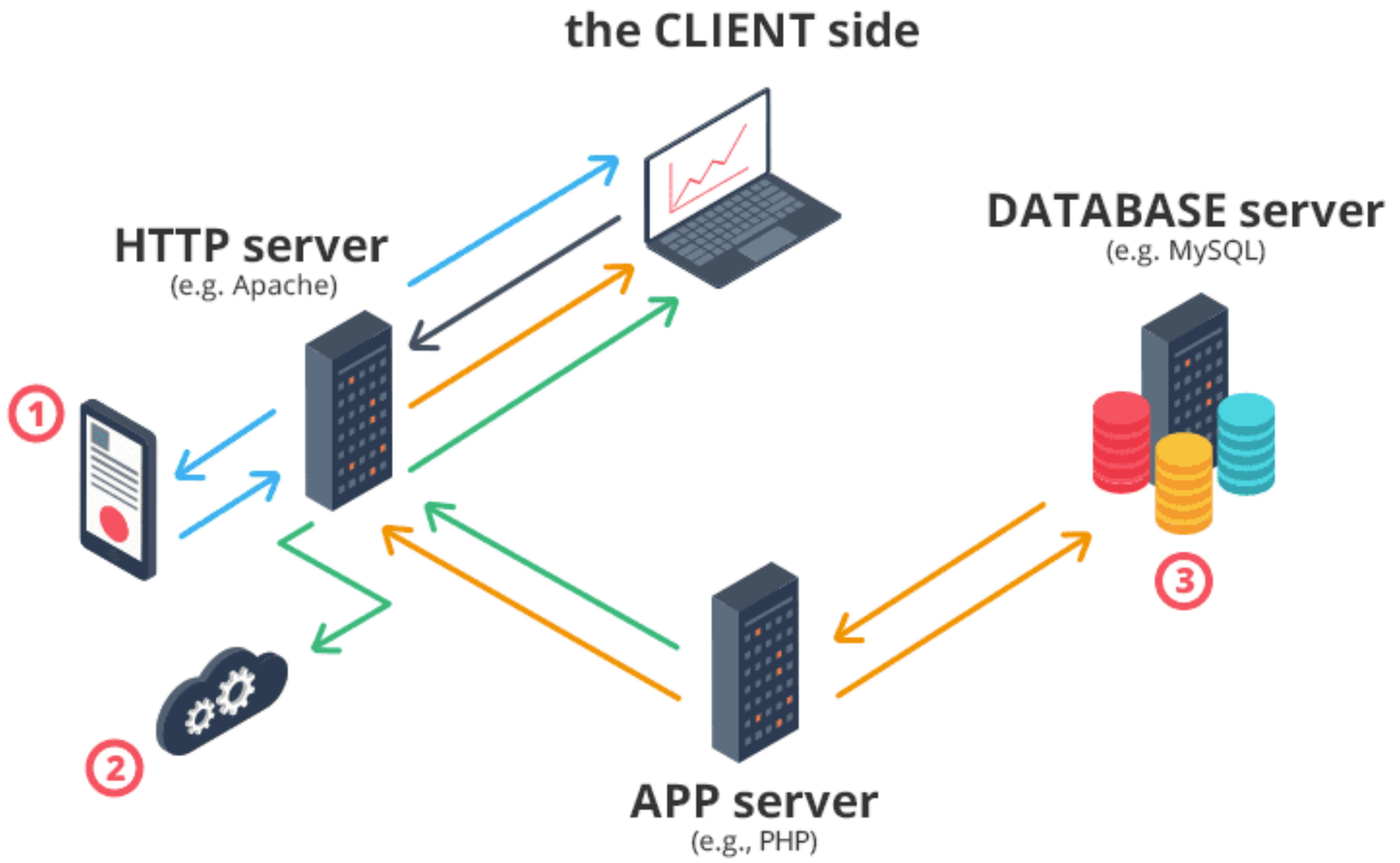
D **Lack of Standardization**

CredChain: In One Slide

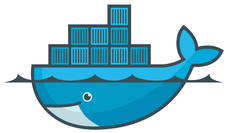
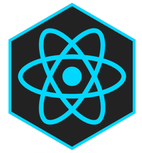
- To address the mentioned challenges, we are proposing ***CredChain***, a next-generation Web3 platform that will revolutionize the verification of academic and professional credentials with the help of blockchain technology (i.e. Algorand).

- **Key Features**
  - **One-Click Verification:** Employers can instantly verify credentials with cryptographic certainty.
  - **Tamper-Proof Records:** All credentials are immutably stored on Algorand blockchain.
  - **Time-Bound Access:** Smart contracts automatically revoke access after verification period ends.
  - **Selective Disclosure:** Candidates control exactly what information is shared with each verifier.
  - **Real-Time Verification:** No waiting periods for institutional responses.
  - **Audit Trail:** Complete history of credential issuance and verification.

Data Flow - In One Slide



Essential Tech-Stack For Our Project

Blockchain	 Algorand	 Docker	 Pera Wallet	 Python
Front-End	 HTML	 CSS	 JavaScript	 React
Back-End	 Node.Js	 Typescript	 Express.Js	
Database	 MongoDB	 IPFS		
Servers	 AWS	 Server	 Local Host	

Module Implemented & Details

User Portal Module

1. Document upload by User
2. Document from DigiLocker
3. Documents from Third Parties (University)

1. Document Self Uploaded

Start

User Login using credentials and goto dashboard.

Upload Document

User will upload document & certificates from their dashboard.

Sign Document & Mint NFT (ASA)

User will sign documents and mint NFTs (ASA) for documents.

Company Validates

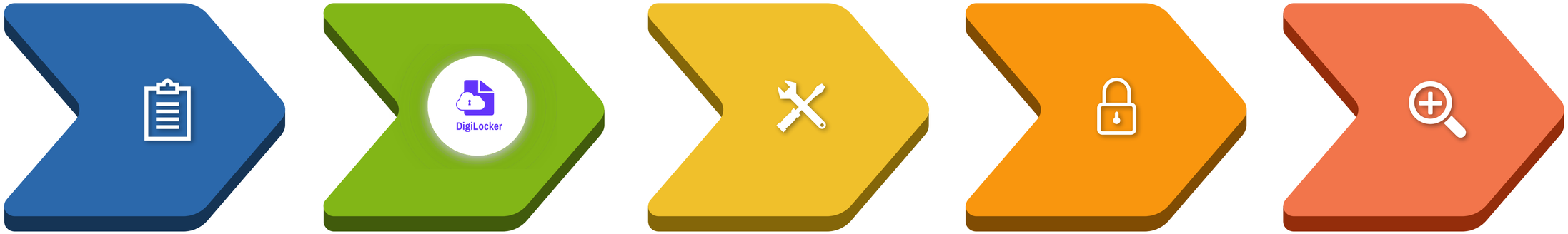
Companies will validates this document. Human intervention needed.

Documents ready to Share

After validation Documents are ready to share.

2. Document from DigiLocker

Note: Real validation using the Digilocker API is outside the scope of this project, and we have been mocking it with our own API.



Start

User Login using credentials and goto dashboard.

Connect to DigiLocker

User will connect to DigiLocker.

Fetch Documents to Database

Fetch document from Digilocker and store hash of metadata in Database.

Sign & Hash Document

Sign and Hash and store for future use.

No validation Required & Docs are ready to Share

No further validation are required for the document from digilocker and Documents are ready to share

3. Documents from Third Parties 3P (University)

Note: Third-party validation and NFT creation are outside the scope of this project and have been hardcoded.

Start & User Requests

User Login using credentials and goto dashboard.

Authorise & Conversion NFT(ASA)

document & certificates from their dashboard.

Sign & Hash Document

User will sign and hash the document using private key.

No validation & 3P Sends Docs to User Address

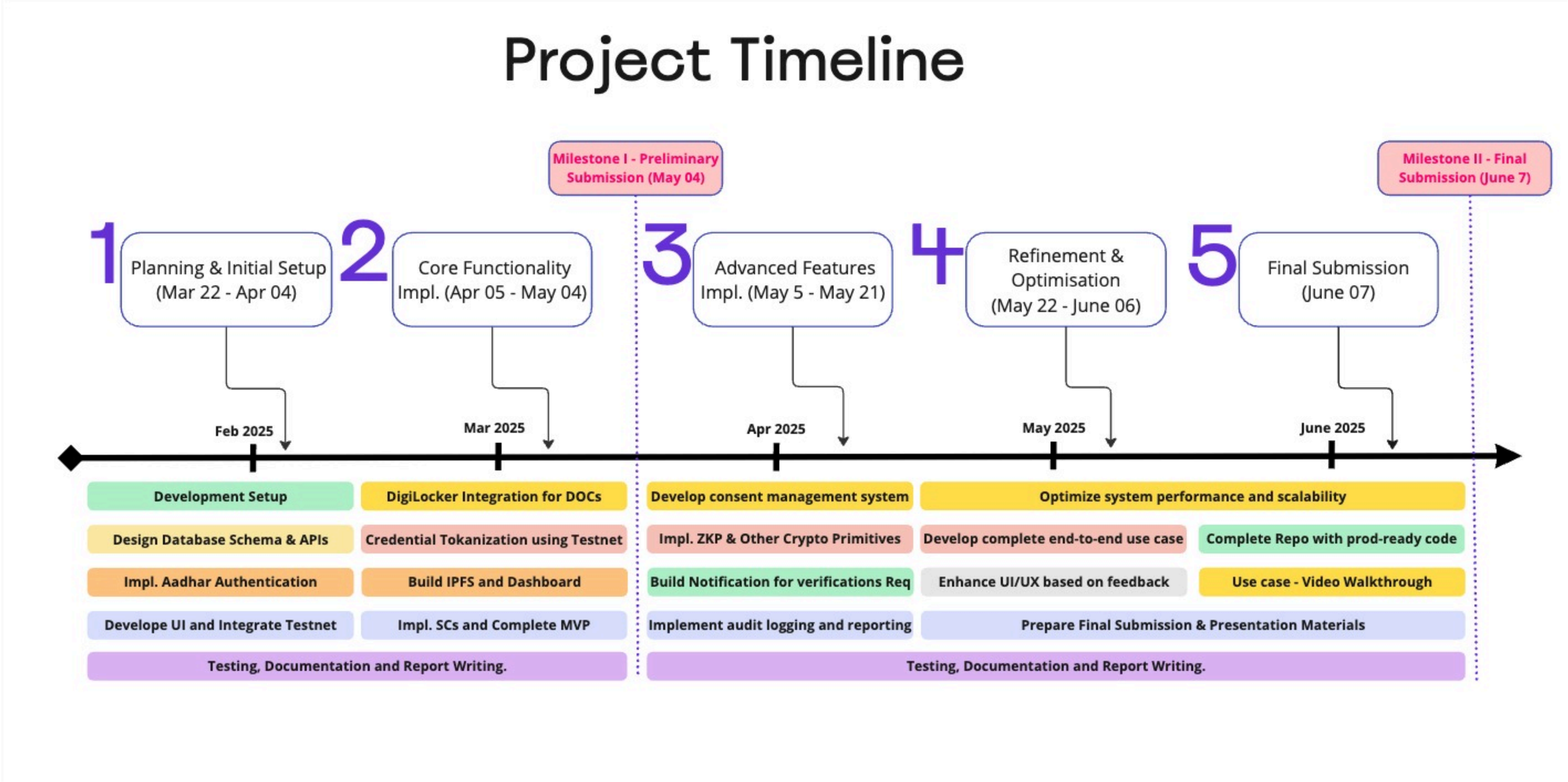
No further validation are required & 3P sends docs to user wallet address.

User Receives Digital Asset → User Signs & Gets New Hash (Save in DB)

User get docs as NFT (ASA) and sign and get new hash for docs and store it in Database for future use.

Demo & Code Walkthrough

Concluding Remarks



Thank You