Decentralised Online Social Network

Major Project

Guide: Prof. Alfred Thomas

Abhiram S (KTE20CS002)
Abhishek Raymond (KTE20CS003)
Abin Augustine (KTE20CS063)
Visakh Vijay O (KTE20CS059)

Overview

Introduction

This project aims to develop a decentralized social networking platform by using blockchain and ipfs technology. The platform provides users with a secure, transparent, and user-centric online social experience. By decentralizing data storage and user interactions, it enhances user control, and data ownership, thereby offering a more empowering and trust-based social network.

Literature Survey

Title	Year	Description
A decentralized social network architecture [1]	2021	This study presents the development of a decentralized social network architecture, incorporating essential social media functionalities, with a primary focus on safeguarding user data privacy
An Enhanced Decentralized Social Network based on Web3 and IPFS using Blockchain [2]	2023	This study proposes a system where the Web3 Library is used to implement the decentralized network. This provides a social graph with all devices in the network.
Performance Evaluation of Decentralized Social Media on Near Protocol Blockchain[3]	2023	This study aims to demonstrate and evaluate the performance of decentralized social media in terms of throughput and scalability.
D-Space: A Decentralized Social Media App[4]	2023	This research aims to explore the various motivations for a decentralized approach to online social networking and the challenges and opportunities associated with decentralization.

Gap Analysis

- No mechanism for content moderation
- Lack of messaging functionality
- Difficult to use for regular users
- Limited user adoption

Problem Statement

To design, and develop a decentralized social networking platform that empowers users by resisting content censorship, and providing robust data ownership. We aim to address our concerns about content censorship and monopolistic control of user data necessitating more open and democratic alternatives.

Proposed System

A decentralized web-based social network prioritizing data ownership, and censorship resistance, containing mechanisms for content moderation, messaging functionality, chronological news feed, and decentralized content sharing.

Requirement Analysis

The requirements were gathered through:

- Literature survey
- Study of existing systems and gap analysis
- User observations

Functional Requirements I

- **User Registration and Login:**
  - User authentication should be required to ensure that only registered users can access the social media.
- **Profile Creation and Updation:**
  - After user registration, user can create their profile by filling the profile creation form.
  - Later user can update their profile by updating this form.
- **Content Sharing:**
  - Allow users to share various types of content, such as text, images, and links, with their network.

Functional Requirements II

- **Chronological Feed:**
  - Chronological feed for displaying user-generated content in order of creation, with the most recent content at the top.
- **Decentralized Data Ownership:**
  - Leverage blockchain technology to grant users true ownership and control of their data, including posts, comments, and profile information.
- **Decentralized Chatting:**
  - Allows users to chat with other users in a secure peer-to-peer network.

Functional Requirements III

- **User Reporting and Moderation Tools:**
  - Enable users to report inappropriate or harmful content and provide users with tools to manage and review reported content.
- **User-Friendly Interface:**
  - Intuitive and user-friendly interface for easy navigation and interaction within the platform.

Non-functional Requirements

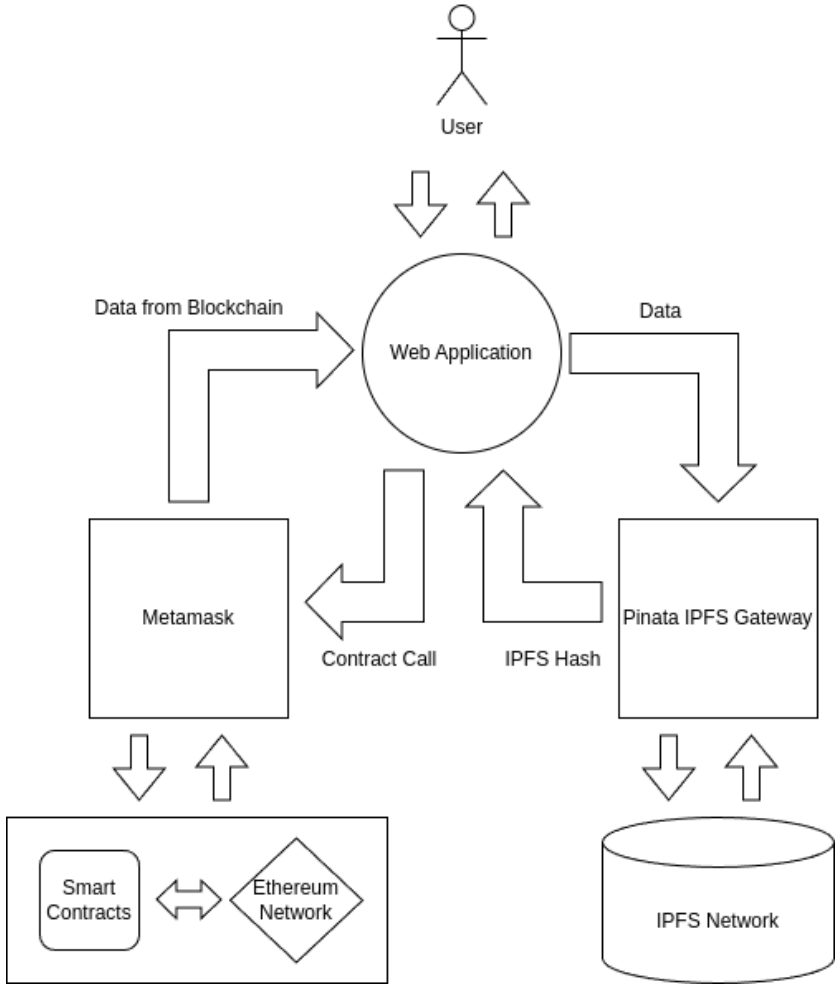
Performance Requirements

- Scalability
- Fast Load Times
- Responsiveness

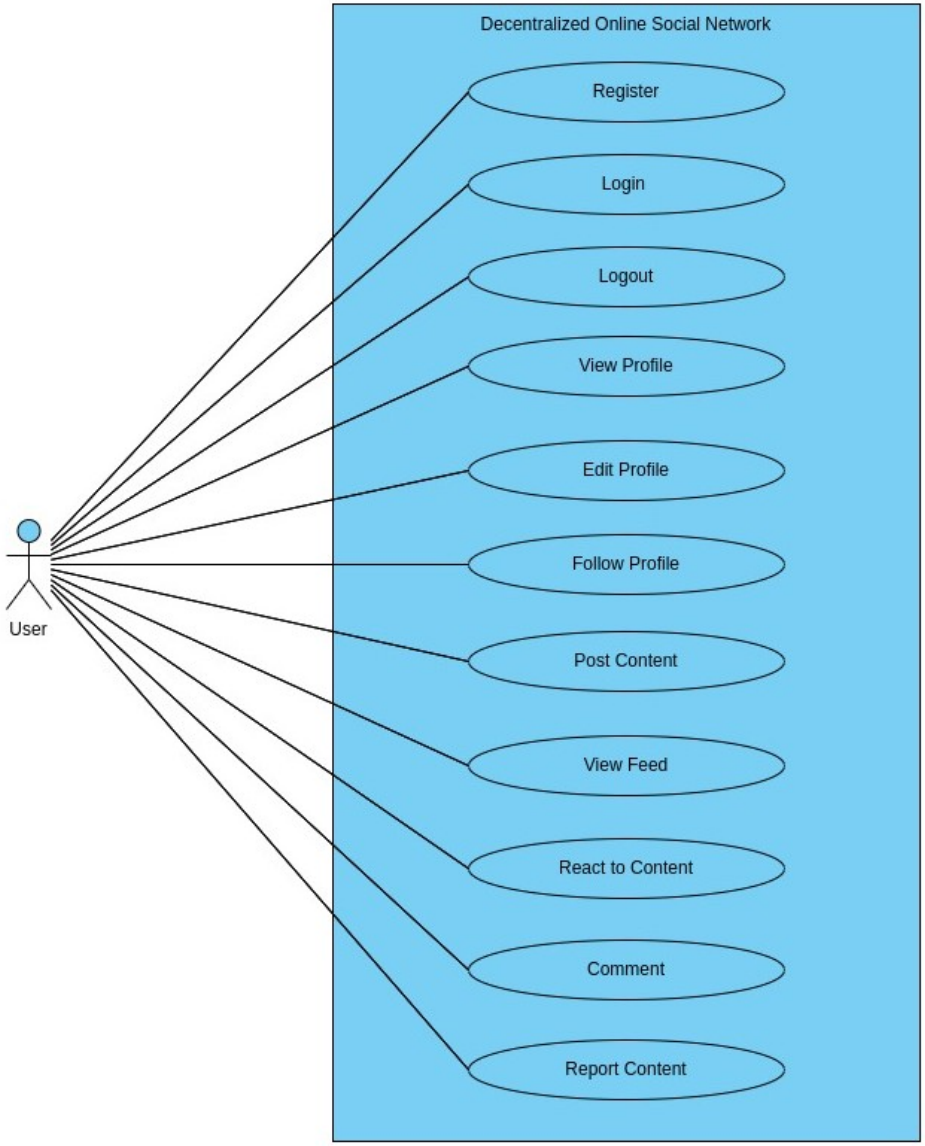
Security Requirements

- Data Security
- Secure User Authentication

System Design



Usecase Diagram



Detailed Design

- Module 1 User Authentication
- Module 2: Account Management
- Module 3: User Interaction
- Module 4: Content Moderation

Account Registration

Algorithm

Algorithm 1: Account Registration

Data: User name U_{name}, Master Secret Key U_{msk}
Result: User account created on the decentralized network with blockchain

(1) $U_{name} \leftarrow$ getUserInput();
(2) $U_{msk} \leftarrow$ getUserSecretKey();
(3) **if** isAlreadyRegistered(U_{msk}) **then**
(4) | alert('User already Exists');
(5) **else**
(6) | registerAccountOnBlockchain(U_{name}, U_{msk});
(7) | navigateTo(Create Profile Page);
(8) **end**

Account Login

Algorithm

Algorithm 2: Account Login

Data: User name U_{name}, Master Secret Key U_{msk}
Result: User logged into the account

```
(1) if !isUserLoggedIn() then
(2)     U_msk ← getUserSecretKey();
(3)     logIntoAccount(U_name, U_msk);
(4) end
```

Profile Creation/Updation

Algorithm

Algorithm 3: Profile Creation/Updation with Blockchain

Data: User Profile Data U_{pd}, Master Secret Key U_{msk}
Result: User Profile is created or Updated on the decentralized network with blockchain

(1) $U_{pd} \leftarrow$ getUserInput();
(2) $U_{msk} \leftarrow$ getUserSecretKey();
(3) **if** isNewUser(U_{msk}) **then**
(4) | createUserProfileOnBlockchain(U_{pd}, U_{msk});
(5) **else**
(6) | updateUserProfileOnBlockchain(U_{pd}, U_{msk});
(7) **end**

Post Upload

Algorithm

Algorithm 4: Post Upload

Data: Master Secret Key U_{msk}, Master Public Key U_{mpk}, Post Content $P_{content}$, Attachment Files $P_{attachments}$

Result: Post is uploaded to the decentralized social network

(1) $U_{msk} \leftarrow$ getUserSecretKey();
(2) $P_{content} \leftarrow$ getPostContent();
(3) $P_{attachments} \leftarrow$ getAttachmentFiles();
(4) $P_{IPFS\_Hash} \leftarrow$ storePostToIPFS($P_{content}$, $P_{attachments}$);
(5) $P_{timestamp} \leftarrow$ getCurrentTimestamp();
(6) $P_{metadata} \leftarrow$ createPostMetadata(U_{mpk},P_{IPFS_Hash}, $P_{timestamp}$);
(7) $P_{id} \leftarrow$ addPostToBlockchain(U_{msk}, $P_{metadata}$);
(8) **if** P_{id} *is not null* **then**
(9) | showSuccessMessage("Post uploaded successfully");
(10) **else**
(11) | showErrorMessage("Failed to add post to blockchain");
(12) **end**

Report Posts

Algorithm

Algorithm 5: Report Posts

Data: User Credentials U_{creds}, Post ID P_{id}
Result: User's report action on a post in the decentralized social network

(1) $U_{creds} \leftarrow$ getUserCredentials();
(2) $U_{msk} \leftarrow$ getUserSecretKey(U_{creds});
(3) **if** *reportButtonIsPressed* **then**
(4) $\quad P_{id} \leftarrow$ getPostID();
(5) $\quad$ **if** isPostNotAlreadyReported(P_{id}, U_{msk}) **then**
(6) $\quad\quad$ addReportPost(P_{id});
(7) $\quad$ **end**
(8) **end**

View and Follow Users

Algorithm

Algorithm 6: View and Follow Users

Data: User Credentials U_{creds}, Target Credentials T_{creds}
Result: User's view and follow actions in the decentralized social network

```
(1)  U_creds ← getUserCredentials();
(2)  T_creds ← getTargetCredentials();
(3)  T_mpk ← getTargetPublicKey(T_creds);
(4)  if T_mpk is not null then
(5)      T_pd ← getTargetProfileFromBlockchain(T_mpk);
(6)      if T_pd is not null then
(7)          showUserProfile(T_pd);
(8)          if followButtonIsPressed && !isUserAlreadyFollowing(T_mpk U_mpk,
               U_target_mpk) then
(9)              addFollowingToBlockchain(U_mpk, U_target_mpk);
(10)         end
(11)     else
(12)         showErrorMessage("Could not fetch target profile");
(13)     end
(14) else
(15)     showErrorMessage("Failed to retrieve target user's public key");
(16) end
```

Like Posts

Algorithm

Algorithm 7: Like Posts

Data: User Credentials U_{creds}, Post ID P_{id}
Result: User's like or unlike action on a post in the decentralized social network

```
(1)  U_creds ← getUserCredentials();
(2)  U_msk ← getUserSecretKey(U_creds);
(3)  if likeButtonIsPressed then
(4)      P_id ← getPostID();
(5)      if isUserAlreadyLiked(P_id) then
(6)          removeLikeFromPost(P_id);
(7)          decrementLikeCount(P_id);
(8)      else
(9)          addLikeToPost(P_id);
(10)         incrementLikeCount(P_id);
(11)     end
(12) end
```

Tools Used

- ReactJs,
- Tailwind CSS
- Ganache
- Metamask
- Truffle
- Pinata(IPFS gateway)
- Polygon Testnet

Future Scope

- Reward for content creation can be given to the users.
- Real-time messaging
- Multiple transactions can be combined into a single event to reduce transaction count.
- Dedicated IPFS server

Limitations

- Scalability
- Loading time issues
- Technical barriers for adoption.

Conclusion

- The decentralized social network ensures content ownership, and resistance to censorship, offering a secure and transparent user-centric online experience.

References I

- Ian Sommerville, Software Engineering, Pearson Education, Tenth edition, 2015.

[1] Tharuka Sarathchandra and Damith Jayawikrama. "A decentralized social network architecture". In: *2021 International Research Conference on Smart Computing and Systems Engineering (SCSE)*. Vol. 4. 2021.

[2] D. Palanikkumar et al. "An Enhanced Decentralized Social Network based on Web3 and IPFS using Blockchain". In: *2023 7th International Conference on Trends in Electronics and Informatics (ICOEI)*. 2023.

[3] Dwynn Tama and Arya Wicaksana. "Performance Evaluation of Decentralized Social Media on Near Protocol Blockchain". In: *2023 17th International Conference on Ubiquitous Information Management and Communication (IMCOM)*. 2023.

[4] Mahavir A. Devmane. "D-Space: A Decentralized Social Media App". In: *2023 2nd International Conference on Edge Computing and Applications (ICECAA)*. 2023.

THANK YOU