SocialChain: A Decentralized Social Media Platform on the Blockchain

Mallellu Sai Prashanth[1(✉)], V. Uma Maheswari[2], Rajinikanth Aluvalu[3], and M. V. V. Prasad Kantipudi[4]

[1] Departmentof Computer Science and Engineering, Vardhaman College of Engineering, Hyderabad, India
saiprashanth08@ieee.org

[2] Departmentof Computer Science and Engineering, Chaitanya Bharathi Institute of Technology, Hyderabad, India

[3] Departmentof Information Technology, Chaitanya Bharathi Institute of Technology, Hyderabad, India
rajanikanth.aluvalu@ieee.org

[4] Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, India
prasad.kantipudi@sitpune.edu.in

Abstract. The emergence of blockchain technology has paved the way for innovative solutions in various domains, including social media. This research paper presents the design and implementation of "SocialChain," a decentralized social media platform built on the Ethereum blockchain using the Solidity programming language. SocialChain aims to address issues related to data privacy, content ownership, and censorship prevalent in traditional social media platforms. The proposed system leverages blockchain's inherent properties such as immutability, transparency, and decentralization to create a trustless and secure environment for social interactions. The platform allows users to create posts, share content, and engage in discussions while maintaining ownership and control over their data. The underlying blockchain ensures that the history of posts and interactions remains tamper-proof and verifiable. The core components of the SocialChain smart contract include data structures for posts and comments, functions for creating posts and adding comments, as well as mechanisms for retrieving post-related information. Each user's identity is associated with their Ethereum address, eliminating the need for traditional authentication methods and enhancing user pseudonymity. To ensure efficiency, the contract optimizes data storage by using mappings and arrays, minimizing gas costs and improving scalability. The contract also incorporates modifiers to validate post and comment existence, enhancing security and preventing unauthorized access. The results of this research showcase the feasibility of integrating blockchain technology into social media platforms. However, it is important to note that while the implementation provides a foundational framework, practical deployment requires addressing challenges such as user experience, scalability, and incentivization mechanisms. In conclusion, SocialChain exemplifies the potential of blockchain technology to revolutionize the landscape of social media by offering a decentralized, secure, and transparent platform for communication and content sharing. This research contributes to the growing body

O. Castillo et al. (Eds.): PerSOM 2023, LNICST 517, pp. 203–219, 2024.
https://doi.org/10.1007/978-3-031-66044-3_14

of knowledge in blockchain applications and encourages further exploration in the realm of decentralized social networking.

Keywords: Social Media · Hyper Ledger · Smart Contracts · Social Chain

1 Introduction

Social media platforms have become an integral part of modern life, transforming the way we communicate, connect, and share information. These digital landscapes have revolutionized the concept of social interaction, enabling individuals, communities, and businesses to engage on a global scale. Social media encompasses a wide range of online platforms and applications that facilitate the creation, sharing, and exchange of content, ideas, and opinions. At the core of social media's appeal is its ability to bridge geographical distances and connect people from diverse backgrounds. Whether it's sharing a personal milestone, discussing current events, or exploring niche interests, social media provides a virtual space where users can express themselves, find like-minded individuals, and build networks that transcend physical boundaries. This has led to the rise of digital communities centered around hobbies, causes, professions, and cultural identities [1]. One of the defining features of social media platforms is user-generated content. Instead of being passive consumers of information, users actively contribute by posting text, images, videos, and other media. This democratization of content creation has empowered individuals to become content creators and influencers, reshaping the dynamics of information dissemination. However, it has also raised questions about the accuracy and credibility of information shared on these platforms, leading to discussions about misinformation, fake news, and content moderation. As social media evolved, it gave birth to various formats and functions, catering to different communication needs. Microblogging platforms like Twitter enable concise updates, while visual platforms like Instagram and Pinterest emphasize images and aesthetics. Video-sharing platforms like YouTube and TikTok revolutionized how we consume entertainment and educational content. Additionally, professional networks like LinkedIn and interest-based communities like Reddit offer specialized spaces for networking and discussions [2].

While social media has connected people like never before, it has also prompted discussions about privacy, data security, and mental health. The vast amount of personal data shared on these platforms has led to concerns about user privacy and the potential misuse of information. Furthermore, the constant exposure to curated content and the pressure to present an idealized version of oneself have raised concerns about the impact of social media on mental well-being. Businesses and brands have also recognized the immense potential of social media for marketing and customer engagement. Social media marketing strategies leverage the platforms' wide reach and targeting capabilities to promote products, services, and campaigns directly to specific audiences. Social media influencers, who have amassed substantial followings, play a significant role in shaping consumer behavior and brand perception [3]. As the landscape of social media continues to evolve, new technologies like blockchain are being explored to address issues of transparency, data ownership, and content monetization. The integration of

blockchain technology could potentially offer solutions for verifying the authenticity of content, protecting user data, and creating decentralized platforms where users have more control over their interactions and contributions [4]. A groundbreaking innovation at the intersection of technology and communication, the concept of a "Social Media Platform on the Blockchain" has emerged as a transformative solution to the challenges and concerns that have long plagued traditional social networking. This innovative approach combines the decentralized and tamper-proof nature of blockchain technology with the ubiquitous realm of social media, ushering in a new era of privacy, security, ownership, and trust for online interactions [8]. At its core, a Social Media Platform on the Blockchain reimagines the conventional social networking landscape by placing users firmly in control of their data and content. Unlike centralized platforms that accumulate vast amounts of personal information and wield significant influence over user interactions, a blockchain- powered social media ecosystem provides users with the sovereignty to manage their profiles, posts, and interactions autonomously. Each piece of content, whether a post, comment, or multimedia file, is securely stored within the blockchain, making it virtually immutable and resistant to unauthorized alterations [5]. One of the most noteworthy advantages of this paradigm shift is the heightened level of data privacy and security it affords. Users can interact with confidence, knowing that their personal information is encrypted, and their interactions are verifiable without requiring intermediaries. This safeguards against data breaches and the exploitation of user information for targeted advertising or other nefarious purposes. The decentralization inherent in the blockchain structure eliminates the risk of a single point of failure, reducing the likelihood of widespread data leaks and privacy infringements [6]. Moreover, a Social Media Platform on the Blockchain introduces a novel approach to content ownership and monetization. With the integration of non-fungible tokens (NFTs), users can attach ownership rights to their posts and creative works, enabling them to monetize their contributions directly. This shift holds immense promise for content creators, as they can receive fair compensation for their work without the need for intermediaries or convoluted monetization schemes [7].

2 Related Work

Blockchain-Based Social Media Platforms

In recent years, traditional centralized social media platforms have faced increasing scrutiny for issues related to data privacy, content manipulation, and lack of user control. In response to these challenges, blockchain technology has emerged as a promising solution to revolutionize the way we interact on social media. Blockchain-based social media platforms offer decentralized content sharing, tokenization, and incentives, fostering a new era of user-centric, secure, and rewarding online interactions [11].

3 Decentralized Content Sharing

One of the primary benefits of blockchain-based social media platforms is their ability to provide users with true ownership and control over their data and content. In traditional platforms, user-generated content is stored on centralized servers, leaving it susceptible

to censorship, alteration, and unauthorized access. In contrast, blockchain technology enables content to be distributed across a decentralized network of nodes, ensuring that each user retains ownership of their data [12]. Users can publish and share content directly on the blockchain, guaranteeing immutability and traceability. Once published, content cannot be altered or removed without consensus from the network, providing transparency and trustworthiness. This architecture empowers users to have full control over their online presence, fostering a sense of autonomy and security [13].

4 Tokenization and Incentives

Blockchain-based social media platforms introduce the concept of tokenization, where users are rewarded with tokens for their engagement, content creation, and participation within the platform. These tokens, often built on blockchain standards like ERC-20 or ERC-721, hold intrinsic value and can be traded, used for various platform services, or converted into other cryptocurrencies [14]. Tokenization incentivizes users to contribute high-quality content, engage with others, and foster meaningful interactions. As users accumulate tokens, they become stakeholders in the platform's ecosystem, aligning their interests with the platform's success. This incentive model creates a positive feedback loop, encouraging users to actively participate while also enhancing the overall user experience [15].

5 Proposed Work

The algorithm for the proposed model is as follows:

Algorithm:

1. Initialize contract state:
   - Define data structures for posts and comments.
   - Maintain a post count and user-to-post mapping.
2. Function createPost(_content: string):
   - Increment postCount.
   - Store the new post:
   - Create a Post struct with postCount, msg.sender, _content, and block.timestamp.
   - Add the Post struct to the 'posts' mapping.
   - Append the post ID to the user's post list.
   - Emit a 'PostCreated' event with post ID, author, and timestamp.
3. Function addComment(_postId: uint256, _content: string):
   - Ensure that the specified post exists in 'posts'.
   - Determine the next comment ID.
   - Store the new comment:
   - Create a Comment struct with comment ID, _postId, msg.sender, _content, and block.timestamp.

   - Add the Comment struct to the 'postComments' mapping under the respective post.
   - Emit a 'CommentAdded' event with post ID, comment ID, author, and timestamp.
4. Function getPostCount() returns uint256:
   - Return the current postCount.
5. Function getUserPostCount(_user: address) returns uint256:
   - Return the length of the user's post list.
6. Function getUserPost(_user: address, _index: uint256) returns (uint256 postId, string content, uint256 timestamp):
   - Retrieve the post ID from the user's post list at the specified index.
   - Return the post's ID, content, and timestamp from 'posts' mapping.
7. Function getPost(_postId: uint256) returns (address author, string content, uint256 timestamp):
   - Retrieve and return the post's author, content, and timestamp from 'posts' mapping.
8. Function getComments(_postId: uint256) returns Comment []:
   - Return the array of comments associated with the specified post from 'postComments' mapping.
9. Event PostCreated(uint256 indexed postId, address indexed author, uint256 timestamp):
   - Log the creation of a new post with the post ID, author's address, and timestamp.
10. Event CommentAdded(uint256 indexed postId, uint256 indexed commentId, address indexed author, uint256 timestamp):
    - Log the addition of a new comment to a post with the post ID, comment ID, author's address, and timestamp (Fig. 1).

The process model consists of the following functions:

6 CreatePost()

The createPost function within the Solidity code plays a fundamental role in enabling users of the "SocialChain" decentralized social media platform to create and share their content on the blockchain. This function serves as the entry point for users to generate new posts, contributing to the growth and engagement of the platform.When a user invokes the createPost function, they provide a string parameter _content representing the textual content of the post they intend to share. This content could be a text-based message, an image URL, or any other form of media that can be represented by a string. By accepting this content as input, the function accommodates various types of posts, fostering a diverse and engaging user experience.

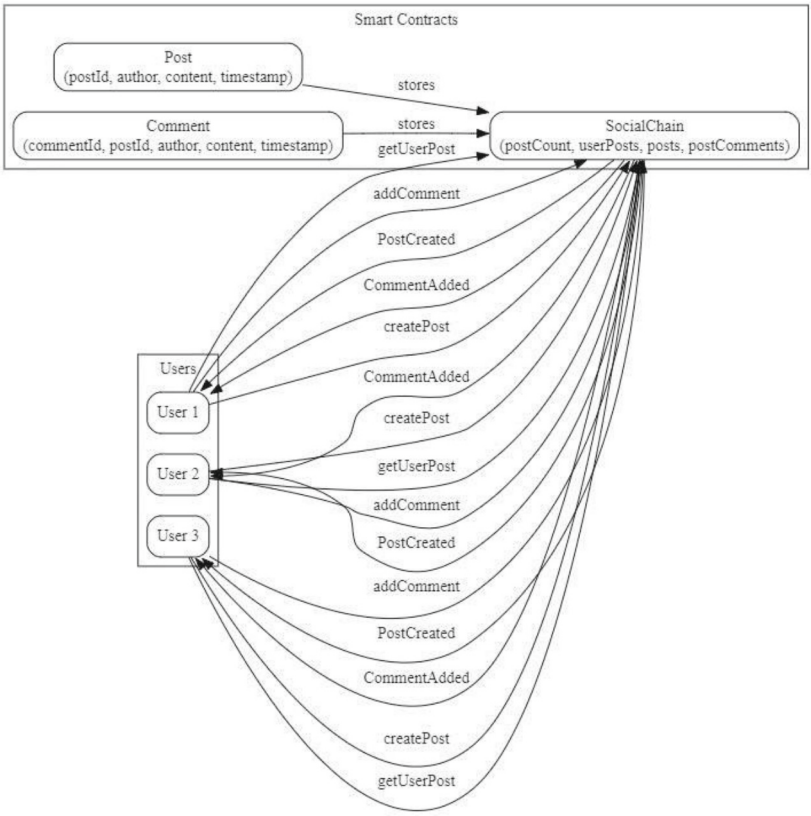

Fig. 1. Architecture diagram of decentralized Social Media Platform

Within the function, a new Post struct is instantiated. This struct encapsulates essential information about the post, including a unique postId identifier assigned incrementally, the Ethereum address of the post's author, the content of the post, and a timestamp denoting when the post was created. This timestamp is captured using the block.timestamp Solidity feature, ensuring the recording of accurate creation times. The newly created Post struct is then stored in the posts mapping, using the postCount as the index. This index-based storage structure facilitates efficient retrieval of posts by their identifiers. Additionally, the function appends the postId to an array stored in the userPosts mapping associated with the author's Ethereum address. This array maintains a record of the posts authored by each user, facilitating easy access to their own content.

Subsequently, an PostCreated event is emitted, allowing external applications and users to listen for new post creation events on the blockchain. This event includes relevant details such as the postId, the author's Ethereum address, and the timestamp of creation. Emitting events like this enhances the transparency and auditability of the platform by making key interactions observable on the blockchain. In summary, the createPost function serves as a pivotal mechanism for users to contribute to the "SocialChain" platform's content ecosystem. It enables the generation of new posts, ensures proper

storage and indexing of post data, and provides an event-driven architecture for external parties to interact with and respond to newly created content. Through this function, users can participate in the creation and sharing of diverse content while leveraging the security and immutability of the blockchain.

7 addComment()

The addComment function in the provided Solidity code is a crucial component of the decentralized social media platform. This function is responsible for allowing users to engage with posts by adding comments to them. Comments are an integral part of any social media platform, enabling users to share their thoughts, opinions, and feedback on the content shared by others. By implementing the addComment function, the contract facilitates interaction and collaboration within the platform's decentralized ecosystem. When a user wishes to add a comment to a specific post, they call the addComment function and provide two essential pieces of information: the _postId of the post they are commenting on, and the _content of the comment they want to contribute. The _postId serves as an identifier for the post to which the comment is being added, ensuring that the comment is associated with the correct content.

Upon receiving these parameters, the function processes the information and performs a series of actions. It first verifies that the provided _postId corresponds to an existing post within the system. This validation ensures that users cannot add comments to non-existent or deleted posts, enhancing the integrity of the platform. Once the post's existence is confirmed, the function generates a unique commentId for the comment being added. This commentId is crucial for tracking and referencing the comment in the future. The function then constructs a Comment struct, comprising the commentId, _postId, the address of the comment's author (the user who called the function), the actual _content of the comment, and a timestamp indicating when the comment was added.

The constructed comment is subsequently appended to the array of comments associated with the specified post. This array, referred to as postComments, stores all comments related to a particular post, allowing users to retrieve and display comments for each post. Additionally, emitting the CommentAdded event provides a mechanism for external systems and user interfaces to react to the addition of a new comment, enhancing the real-time interaction experience. In summary, the addComment function is a critical part of the decentralized social media platform, fostering engagement and interaction among users. It ensures that comments are securely associated with the correct posts, providing a seamless and transparent experience for users who wish to express their opinions, initiate discussions, and actively participate in the platform's community. Through this function, the platform creates an environment where decentralized interactions are facilitated by the power of blockchain technology.

8 getPostCount()

The getPostCount() function plays a crucial role within the Solidity code of the decentralized social media platform. This function is designed to provide users and external applications with a reliable and accurate count of the total number of posts that have

been created on the platform. In the context of SocialChain, which aims to establish a decentralized environment for social media interactions on the blockchain, this function contributes to the transparency and accessibility of platform statistics. When a user or an external entity invokes the getPostCount() function, the contract performs a read-only operation to retrieve the count of posts that have been created since the platform's inception. This count is maintained within the contract's internal storage, and it accurately reflects the current state of the platform.

The significance of the getPostCount() function extends beyond providing a simple numeric value. It serves as a fundamental metric to gauge the level of activity and engagement on the SocialChain platform. Users, developers, and stakeholders can utilize this function to monitor the platform's growth over time. This functionality aligns with the decentralized nature of the blockchain, as it ensures that the count of posts is consistently available to anyone who queries the contract without relying on a centralized authority. Furthermore, the getPostCount() function showcases the efficiency and speed of querying data on the blockchain. Since this function is marked as view, it signifies that it does not modify the state of the blockchain and operates solely by retrieving data from the contract's storage. As a result, users can quickly obtain an up-to-date count of posts without incurring any transaction costs or waiting for confirmations.

In essence, the getPostCount() function exemplifies the potential of integrating blockchain technology into social media platforms. It highlights the accessibility of data, the transparency of statistics, and the elimination of intermediaries. By incorporating this function into the Solidity code, the SocialChain project takes a significant step toward establishing a decentralized ecosystem that empowers users and fosters trust through blockchain-based data availability.

9 getUserPostCount()

The getUserPostCount function in the provided Solidity code serves as an essential query mechanism within the decentralized social media platform. This function offers insights into the posting activity of a specific user, allowing users and external applications to retrieve the count of posts created by a particular Ethereum address. In the context of the decentralized social media platform, this function plays a pivotal role in enabling users to gauge their level of engagement and contribution to the platform's content. By providing a user's post count, the function offers a quantifiable metric of a user's activity and participation, reflecting their involvement in discussions, sharing of content, and interactions with others.

From a technical perspective, the getUserPostCount function operates as a view function, which means it doesn't modify the blockchain's state and only retrieves information. This design choice aligns with the Ethereum blockchain's principle of separating read operations from write operations, ensuring the efficiency of the underlying blockchain network. By allowing users to retrieve their own post count, the function empowers them with a user-centric experience. This user-centric approach contributes to a more transparent and accountable ecosystem, as users can readily access information about their own activities without relying on intermediaries or centralized databases.

External applications and services can also leverage the getUserPostCount function to provide enhanced features, such as user analytics and engagement metrics. This

function could be utilized to create leaderboards, reward systems, or personalized recommendations based on a user's posting history. Such features can foster a sense of community, incentivize active participation, and potentially contribute to the growth of the decentralized social media platform. In conclusion, the getUserPostCount function represents a fundamental building block within the decentralized social media platform's smart contract. By offering users the ability to retrieve their post count, the function enhances transparency, encourages user engagement, and provides valuable data for further platform development and user experience improvements.

10 getUserPost()

The getUserPost function in the provided Solidity code is a crucial component of the decentralized social media platform. This function aims to provide users with the ability to retrieve information about their own posts based on their Ethereum address and a specific index. The purpose of this function is to facilitate a personalized view for users, allowing them to see the posts they have created and their associated details. When a user calls the getUserPost function, they need to provide their Ethereum address (_user) and the index of the post they are interested in (_index). The function operates in a view mode, indicating that it does not modify the blockchain state but only returns data.

The function starts by fetching the specific post ID associated with the user's address and the given index. This is achieved by accessing the userPosts mapping, which maintains an array of post IDs for each user. By using the user's address as the key, the function retrieves the array of post IDs that the user has created. Once the post ID is obtained, the function accesses the posts mapping, which stores detailed information about each post. The relevant information retrieved includes the content of the post and the timestamp when it was created.

By providing users with access to their own posts, the getUserPost function enhances the user experience by allowing them to review their contributions to the social media platform. This function's functionality can be integrated into the platform's user interface, enabling users to navigate through their posts, view content, and engage with their own contributions. It's important to note that while this function is designed to provide users with a personalized view of their posts, the broader context of the decentralized social media platform involves various other functions, including creating posts, adding comments, and fetching post-related information. The seamless integration of these functions creates a comprehensive and user- friendly social media experience within a blockchain ecosystem.

11 getPost()

The getPost function in the provided Solidity code is a crucial component of the decentralized social media platform. This function serves as an interface to retrieve detailed information about a specific post using its unique post ID. The purpose of this function is to allow users and external applications to access important details of a post stored on the blockchain. When a user or application calls the getPost function with a specific _postId parameter, the function first retrieves the essential information associated with

the post from the posts mapping within the contract. The details include the address of the post's author, the content of the post, and the timestamp indicating when the post was created.

This information is then packaged into a tuple and returned to the caller, providing them with a comprehensive snapshot of the post's attributes. By utilizing this function, users can access posts from the blockchain without having to delve into the underlying complexities of the contract's data storage and retrieval mechanisms. The getPost function contributes to the platform's transparency and user-friendliness. Users can easily access and display post content, authorship, and timestamps through their preferred user interfaces, such as web applications or mobile apps. Additionally, the function's external view visibility modifier ensures that it does not modify the contract's state, making it gas-efficient and suitable for querying blockchain data without incurring transaction costs.

It's important to note that while the getPost function provides a convenient way to retrieve post information, its implementation is just one part of the broader decentralized social media platform. Other functions, such as createPost and addComment, complement this retrieval mechanism by enabling the creation and interaction with posts and comments on the platform. Together, these functions form the foundation of a functional, blockchain-based social media ecosystem.

12 getComments()

The getComments function plays a crucial role in the decentralized social media platform implemented in the provided Solidity code. This function serves as a means for retrieving comments associated with a particular post. In the context of a decentralized social media platform, user engagement and interaction through comments are essential components, and this function facilitates the retrieval of these interactions from the blockchain. When a user wishes to view the comments for a specific post, they can call the getComments function and provide the _postId as an argument. The _postId acts as a unique identifier for the target post. Leveraging this identifier, the function accesses the stored comments related to that particular post within the blockchain's storage.

Upon successful execution of the function, it returns an array of comments. Each comment in the array is represented as a structure with several properties, such as commentId, postId, author, content, and timestamp. The commentId is a sequential identifier for the comment within the context of the specific post. The postId denotes the post to which the comment belongs. The author represents the Ethereum address of the user who posted the comment. The content holds the textual content of the comment, and the timestamp indicates when the comment was added. By providing a comprehensive array of comments along with their associated details, the getComments function empowers users to engage with the content on the platform more meaningfully. This function enhances transparency, accountability, and immutability by utilizing the underlying blockchain's properties. All comments are stored in a decentralized and tamper-resistant manner, ensuring that the integrity of the interactions remains intact over time.

From a user's perspective, the ability to access comments through this function fosters an enriched user experience, enabling them to engage in discussions, provide feedback,

and share their thoughts on the content posted by others. The getComments function, coupled with other functionalities provided by the smart contract, forms the backbone of the decentralized social media platform's interaction layer, creating a dynamic and engaging ecosystem for users to connect, share, and interact securely.

13 Results

Trust is a cornerstone of social media interactions, and blockchain technology serves as a bedrock for fostering genuine trust within the digital realm. The transparent and auditable nature of the blockchain ensures that interactions are traceable and authentic, curbing the spread of fake news and malicious content. Additionally, the decentralized nature of the platform mitigates the undue influence of algorithms and the manipulation of user feeds, promoting a more organic and meaningful exchange of ideas [9]. However, the development of a Social Media Platform on the Blockchain is not without its challenges. Scalability, user experience, and regulatory considerations remain focal points for ongoing refinement. Balancing the decentralized ethos of blockchain with the user-friendly features expected from social media platforms requires intricate engineering and design [10] (Figs. 2, 3, 4, 5, 6, 7, 8 and 9).

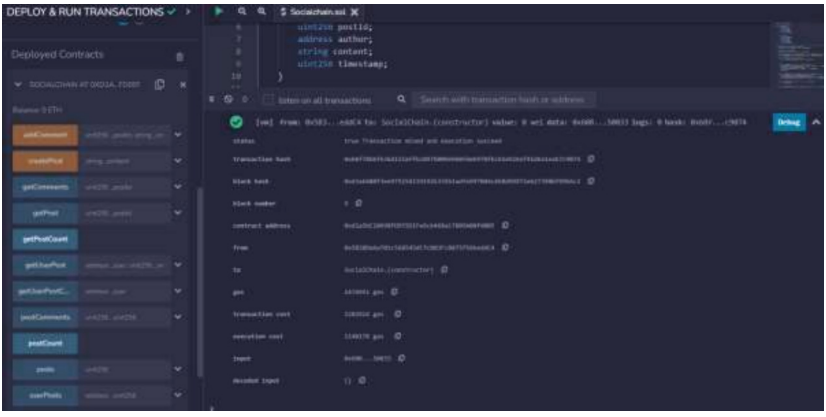

Fig. 2. User interface demonstrating the deployed contract

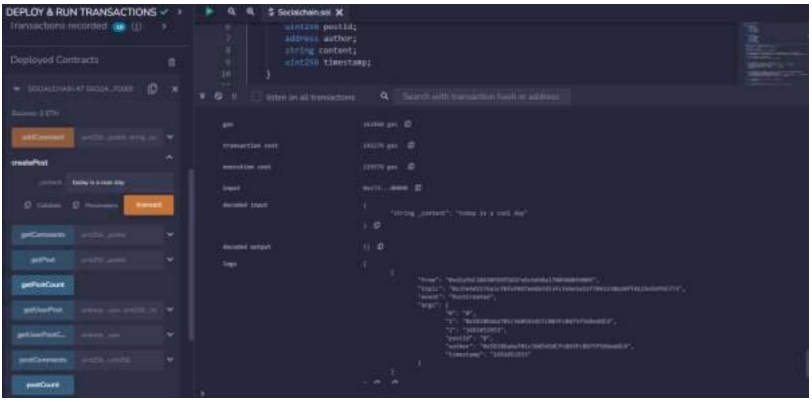

Fig. 3. User interface demonstrating the createPost function

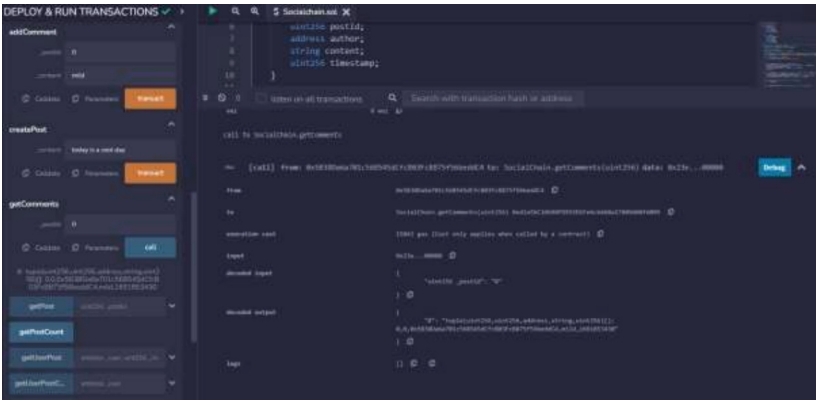

Fig. 4. User interface demonstrating the getComments function

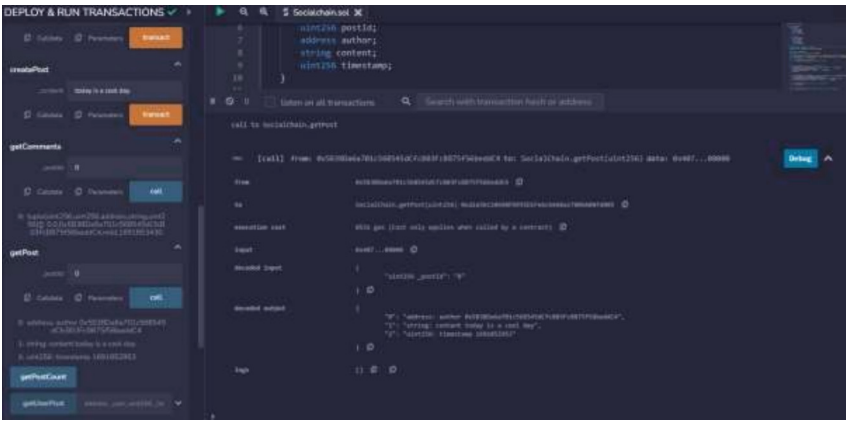

Fig. 5. User interface demonstrating the getPost function

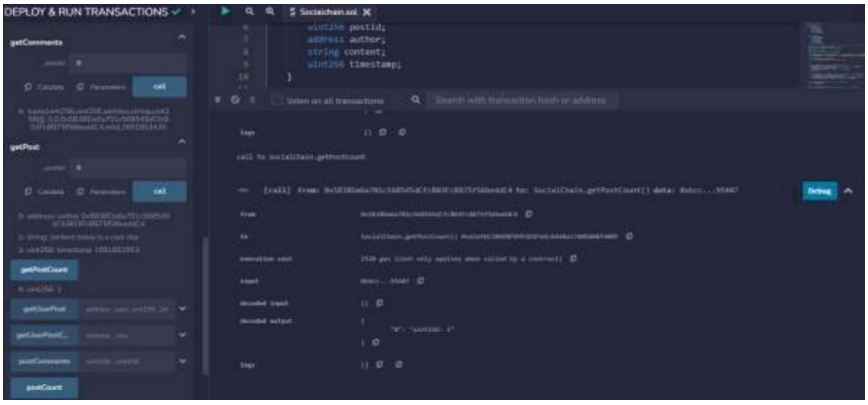

Fig. 6. User interface demonstrating the getPostCount function

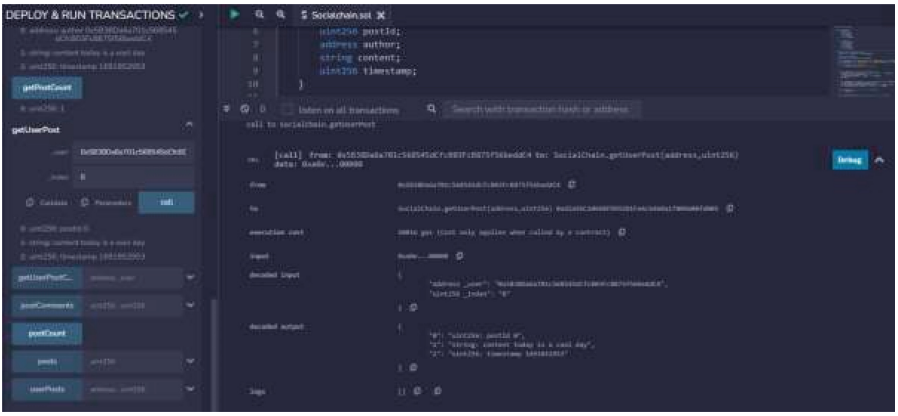

Fig. 7. User interface demonstrating the getUserPost function

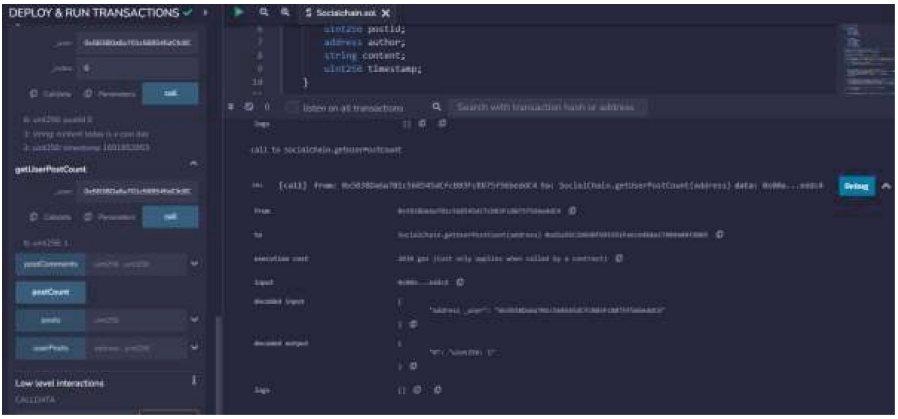

Fig. 8. User interface demonstrating the getUserPostCount function

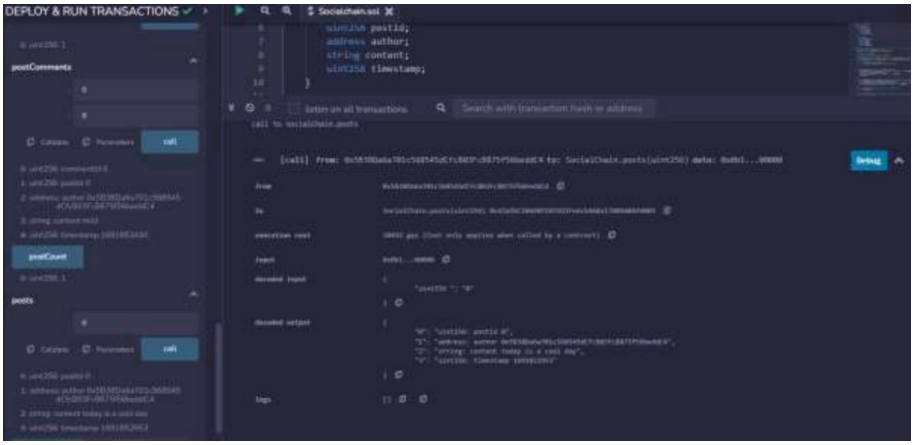

Fig. 9. User interface demonstrating the posts function

14 Conclusion

In conclusion, the presented Solidity code outlines the foundational framework for "SocialChain," a decentralized social media platform implemented on the blockchain. The code provides a simplified yet functional illustration of the essential components required for a social media platform, including post creation, comment addition, and data retrieval. However, it is important to acknowledge that this code represents only a preliminary step towards realizing a fully operational and feature-rich decentralized social media ecosystem. The contract architecture leverages Ethereum's Solidity programming language to establish the underlying structure of SocialChain. Users can create posts and add comments, with each post and comment being associated with unique identifiers, authors' addresses, content, and timestamps. The code showcases the utilization of mappings and arrays to efficiently store and manage post and comment data, ensuring optimal data retrieval and user experience. It is vital to recognize the inherent complexity of developing a complete decentralized social media platform. The outlined code serves as a foundation that can be further expanded and enhanced to incorporate essential features such as user authentication, privacy mechanisms, content moderation, tokenization for incentivization, and user interface integration. As the code stands, it lacks comprehensive user management, a graphical user interface, and crucial security measures, which are indispensable in real-world applications. To transform this basic contract into a functional social media platform, extensive development efforts are required. These efforts should encompass refining smart contract logic, implementing robust user authentication mechanisms, designing an intuitive and user-friendly front-end interface, ensuring data privacy, and addressing scalability concerns. Furthermore, thorough testing and auditing are essential to ensure the security and reliability of the platform, particularly when handling sensitive user- generated content and interactions. In summary, the provided Solidity code represents a foundational blueprint for a decentralized social media platform, highlighting the potential of combining blockchain technology and social networking. However, this code is just a starting point in the journey towards building

a comprehensive and secure decentralized social media ecosystem. As the blockchain landscape continues to evolve, the practical implementation of SocialChain necessitates diligent development, continuous innovation, and adherence to best practices to address the multifaceted challenges and opportunities presented by decentralized social networking.

15 Future Scope

In addition to the current implementation, there are several avenues for future development and enhancement of the proposed SocialChain platform on the blockchain. One promising avenue is the integration of a decentralized identity system to enhance user authentication and data privacy. By utilizing self-sovereign identity solutions, users could maintain control over their personal information, granting them the ability to selectively share data with trusted parties while maintaining anonymity where desired. Furthermore, the integration of decentralized content moderation mechanisms could be explored to tackle issues such as spam, inappropriate content, and fake news. Leveraging community-based moderation, where users collectively curate and moderate content, could foster a more democratic and resilient content ecosystem. Tokenization of the platform's engagement metrics could also be considered as a means of incentivizing user participation and quality contributions. The introduction of a native utility token would enable users to be rewarded for their valuable contributions to the platform, fostering a vibrant community. Enhancing the user experience through the implementation of a user-friendly decentralized application (DApp) interface is vital for user adoption. This DApp could provide an intuitive interface for users to interact with the blockchain-based social media features seamlessly. The user experience should prioritize speed and responsiveness to ensure a fluid and engaging platform. To address scalability challenges inherent in blockchain networks, the integration of Layer 2 solutions, such as state channels or sidechains, could be explored. These solutions could significantly improve the platform's throughput and lower transaction costs, thereby facilitating mass adoption. Lastly, the incorporation of non-fungible tokens (NFTs) could revolutionize content ownership and monetization. NFTs could represent unique digital assets, allowing creators to tokenize their content and sell ownership stakes to their audience, transforming the way content creators and consumers interact economically. In conclusion, the proposed SocialChain platform offers a strong foundation for a decentralized social media ecosystem. However, to realize its full potential, future development should encompass areas such as decentralized identity, content moderation, tokenization, user experience optimization, scalability solutions, and NFT integration. By addressing these aspects, the platform can foster a more secure, engaging, and rewarding social media experience, transforming the way users interact and engage with content on the blockchain.

References

1. Guidi, B., Michienzi, A.: Users and bots behaviour analysis in Blockchain Social Media. In: 2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS), Paris, France, pp. 1–8 (2020). https://doi.org/10.1109/SNAMS52053.2020.9336553

2. Boyd, D.M., Ellison, N.B.: Social network sites: definition history and scholarship. J. Comput.-Mediat. Commun. **13**(1), 210–230 (2007)
3. Shae, Z., Tsai, J.: AI blockchain platform for trusting news. In: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, pp. 1610–1619 (2019). https://doi.org/10.1109/ICDCS.2019.00160
4. Phillips, R.C., Gorse, D.: Predicting cryptocurrency price bubbles using social media data and epidemic modelling. In: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA, pp. 1–7 (2017). https://doi.org/10.1109/SSCI.2017.8280809
5. Tee, W.J., Murugesan, R.K.: Trust network, blockchain and evolution in social media to build trust and prevent fake news. In: 2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA), Subang Jaya, Malaysia, pp. 1–6 (2018). https://doi.org/10.1109/ICACCAF.2018.8776822
6. Paul, S., Joy, J.I., Sarker, S., Shakib, H., Ahmed, S., Das, A.K.: Fake news detection in social media using blockchain. In: 2019 7th International Conference on Smart Computing & Communications (ICSCC), Sarawak, Malaysia, pp. 1–5 (2019). https://doi.org/10.1109/ICSCC.2019.8843597
7. Guidi, B., Michienzi, A.: Interaction communities in blockchain online social media. In: 2021 Third International Conference on Blockchain Computing and Applications (BCCA), Tartu, Estonia, pp. 89–96 (2021). https://doi.org/10.1109/BCCA53669.2021.9657049
8. Dwivedi, A.D., Singh, R., Dhall, S., Srivastava, G., Pal, S.K.: Tracing the source of fake news using a scalable blockchain distributed network. In: 2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Delhi, India, pp. 38–43 (2020). https://doi.org/10.1109/MASS50613.2020.00015
9. Guidi, B., Michienzi, A., Ricci, L.: Steem blockchain: mining the inner structure of the graph. IEEE Access **8**, 210251–210266 (2020). https://doi.org/10.1109/ACCESS.2020.3038550
10. Shahbazi, Z., Byun, Y.-C.: Fake media detection based on natural language processing and blockchain approaches. IEEE Access **9**, 128442–128453 (2021). https://doi.org/10.1109/ACCESS.2021.3112607
11. Ahram, T., Sargolzaei, A., Sargolzaei, S., Daniels, J., Amaba, B.: Blockchain technology innovations. In: 2017 IEEE Technology & Engineering Management Conference (TEMSCON), Santa Clara, CA, USA, pp. 137–141 (2017). https://doi.org/10.1109/TEMSCON.2017.7998367
12. Guidi, B., Michienzi, A., Ricci, L.: Assessment of wealth distribution in blockchain online social media. IEEE Trans. Comput. Soc. Syst. **11**, 671–682 (2022). https://doi.org/10.1109/TCSS.2022.3228925
13. Tang, H., Ni, J., Zhang, Y.: Identification and evolutionary analysis of user collusion behavior in blockchain online social medias. IEEE Trans. Comput. Soc. Syst. (2022). https://doi.org/10.1109/TCSS.2022.3215185
14. Guidi, B., Michienzi, A.: Social games and Blockchain: exploring the metaverse of decentral. In: 2022 IEEE 42nd International Conference on Distributed Computing
15. Systems Workshops (ICDCSW), Bologna, Italy, pp. 199–204 (2022). https://doi.org/10.1109/ICDCSW56584.2022.00045
16. Bogusz, C.I., Laurell, C., Sandström, C.: Tracking the digital evolution of entrepreneurial finance: the interplay between crowdfunding, blockchain technologies, cryptocurrencies, and initial coin offerings. IEEE Trans. Eng. Manage. **67**(4), 1099–1108 (2020). https://doi.org/10.1109/TEM.2020.2984032