

BLOCKCHAIN

Devoir – 6

Rapport de Projet : Mini Twitter

Realisé par:

ABDELALI IBN TABET

## Introduction

L'objectif de ce projet était de développer une application Mini Twitter décentralisée basée sur un contrat intelligent Solidity, avec une interface web permettant aux utilisateurs d'interagir avec la blockchain via MetaMask. Ce rapport détaille les étapes de développement, les fonctionnalités, et les modifications effectuées sur le contrat initial.

## 1 - Création et mise en place de l'environnement

### Configuration de l'environnement de développement

- Outils utilisés : Solidity, JavaScript, HTML, Web3.js, MetaMask.
- Déploiement du contrat sur un réseau de test (Sepolia).

### Connexion MetaMask

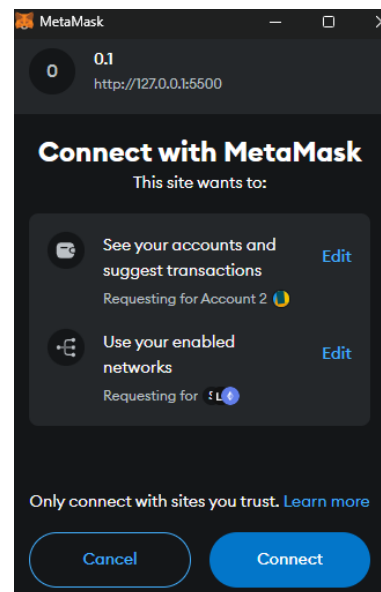
- Ajout d'une fonction pour connecter l'application à un portefeuille MetaMask.
- Fonction ***connectWallet()*** pour se connecter et ***disconnectWallet()*** pour se déconnecter.

## Mini X

Utilisateur non connecté

## Mini X

Connecté en tant que : 0x8ba0b03A05FE7cbE665f3CEd3e195ae013ddA8f6



## 2 - Modification du Contrat MiniSocial.sol

Les modifications du contrat ont été nécessaires pour ajouter de nouvelles fonctionnalités. Voici les changements principaux apportés :

### Ajout de la gestion des *likes/dislikes*

- Ajout des variables likes et dislikes dans la structure Post.
- Création des fonctions ***likePost()*** et ***dislikePost()*** pour enregistrer les likes et dislikes, tout en évitant les doublons par utilisateur.

## Modification des publications par l'auteur uniquement

- Ajout de la fonction ***modifyPost()***, permettant seulement à l'auteur de modifier son post.
- Ajout d'une variable ***lastModified*** pour stocker la date de modification.

## Affichage de la date de création et de la date de modification

- Ajout de ***timestamp*** et ***lastModified*** dans la structure ***Post*** pour afficher la date de publication et, en cas de modification, la date de modification.

## 3 - Création de l'interface JavaScript

### Fonctions pour interagir avec le contrat

- Utilisation de Web3.js pour établir une connexion avec le contrat déployé.
- Fonction ***publishPost()*** pour publier un post et ***loadPosts()*** pour charger les posts depuis la blockchain.

### Ajout de la gestion des likes, dislikes, et modification des posts

- ***likePost(id)*** et ***dislikePost(id)*** sont appelées via des boutons dans l'interface, incrémentant les likes/dislikes pour un post.
- ***modifyPost(id)*** permet à l'auteur de modifier son post en remplissant un prompt JavaScript pour entrer le nouveau message.

### Affichage de la date de publication et de modification

- ***loadPosts()*** affiche chaque post avec la date de création et, si elle est présente, la date de dernière modification.

## Mini X

Se connecter au Wallet

Déconnecter

Connecté en tant que : 0x8ba0b03A05FE7cbE665f3CEd3e195ae013ddA8f6

Nouveau post

Ajouter un post

#FirstPost

Posté par : 0xB1cd5Ca93f24911eA63377019d6366853daF7d30

Date : 11/7/2024, 4:56:36 PM

Modifié : 11/7/2024, 4:59:12 PM

👍 2 | 👎 0

LikeDislike

Bonjour

Posté par : 0x8ba0b03A05FE7cbE665f3CEd3e195ae013ddA8f6

Date : 11/7/2024, 5:01:12 PM

👍 0 | 👎 1

LikeDislikeModifier

Free Palestine 🇵🇸

Posté par : 0xB1cd5Ca93f24911eA63377019d6366853daF7d30

Date : 11/7/2024, 5:07:00 PM

👍 1 | 👎 0

LikeDislike

## Conclusion

Ce projet a permis de créer une mini-application de type Twitter décentralisée, où les utilisateurs peuvent :

- Publier, liker, disliker et modifier leurs posts.
- Voir la date de publication et, si nécessaire, la date de modification.
- Se déconnecter et se reconnecter avec différents comptes MetaMask pour tester l'application sous différents utilisateurs.