

ATELIER 5 : SMART CONTRACT "MINI RESEAU SOCIAL" EN SOLIDITY

Instructions

1. **Déclaration du Smart Contract** : Dans Remix, crée un fichier *MiniSocial.sol* et définis le smart contract MiniSocial. Déclare une structure **Post** qui contiendra un message de type *string* et l'adresse **author** de l'utilisateur qui a publié le message.
2. **Liste des Messages** : Déclare un tableau dynamique *posts* de type **Post[]** pour stocker tous les messages publiés.
3. **Fonction de Publication** : Crée une fonction **publishPost(string memory _message)** qui permet aux utilisateurs de publier un message. Cette fonction doit créer un nouveau Post avec le message et l'adresse de l'utilisateur (*msg.sender*) puis l'ajouter au tableau *posts*.
4. **Fonction de Consultation** : Ajoute une fonction **getPost(uint index) public view returns (string memory, address)** qui prend un index en entrée et retourne le message et l'auteur du post correspondant dans le tableau *posts*.
5. **Fonction pour Récupérer le Nombre de Messages** : Crée une fonction **getTotalPosts()** qui retourne le nombre total de messages publiés.
6. **Déploiement et Tests** : Déploie le smart contract MiniSocial sur un réseau de test. Publie quelques messages en utilisant différents comptes (adresses) et teste la récupération des messages. Décris les étapes pour chaque test et les résultats obtenus.

Outils nécessaires :

- MetaMask (test network : SepoliaETH)
- Remix IDE

Livrable : Il faut mettre le code source dans Github en plus d'un rapport détaillé.