

Projet vote



# Partie 1 :

- ✦ Smart contract de vote :
  - ✦ L'admin du contrat autorise les adresses pouvant proposer des vote et voter
  - ✦ Chaque électeur peut voir les votes de tout le monde
  - ✦ Le gagnant est déterminé par la majorité
  - ✦



# Partie 2

- Déroulement du processus
  - L'admin enregistre une liste d'électeurs par leurs addresses
  - L'admin commence la session d' enregistrements des propositions
  - Les électeurs inscrits enregistrent leurs propositions seulement quand la session d'enregistrement est active
  - L'admin arrête la session d'enregistrement des propositions
  - L'admin commence la session de votes
  - Les électeurs votent sur leur proposition préférée
  - L'admin met fin a la session de vote
  - L'admin lance la comptabilisation des votes
  - Tout le monde peut voir les résultât de la proposition



# Exigences & recommandations

- ✦ Le contrat doit s'appeler Voting
- ✦ On utilise vs code avec la dernière version du compilateur
- ✦ L'admin = déployer du contrat
- ✦ Utilisation de structs et de enums (page suivante)
- ✦ Un uint winning proposalId (l'id de la proposition gagnante) et une fonction getWinner qui retourne l'aid
- ✦ Vous devez utiliser la librairie Ownable de openzeppelin



# Struct et enums à utiliser

```
struct Voter {  
    bool isRegistered;  
    bool hasVoted;  
    uint votedProposalId;  
}  
struct Proposal {  
    string description;  
    uint voteCount;  
}
```

```
enum WorkflowStatus {  
    RegisteringVoters,  
    ProposalsRegistrationStarted,  
    ProposalsRegistrationEnded,  
    VotingSessionStarted,  
    VotingSessionEnded,  
    VotesTallied  
}
```



# En utilisant hardhat sous vs code

- ✦ Créer des scripts js de changement d'état dans le contrat (étapes du vote)
- ✦ Mettre en place une serie de tests unitaires
- ✦