



UNIVERSITÀ DEGLI STUDI  
DI SALERNO

# SecureVoting: Voting made Safer

Anno accademico: 2022 / 2023

Giovanni Rapa e Antonio Cacciapuoti

# Chi siamo



**Antonio  
Cacciapuoti**

Studente magistrale iscritto al  
curriculum di Cloud Computing



**Giovanni Rapa**

Studente magistrale iscritto al  
curriculum di Sicurezza  
Informatica

# Indice

01

## Cos'è?

Breve descrizione e le tecnologie utilizzate per l'implementazione

03

## SecureVoting.sol

Implementazione dello smart contract

02

## Raccolta dei requisiti

Analisi delle esigenze e dei vincoli degli utenti

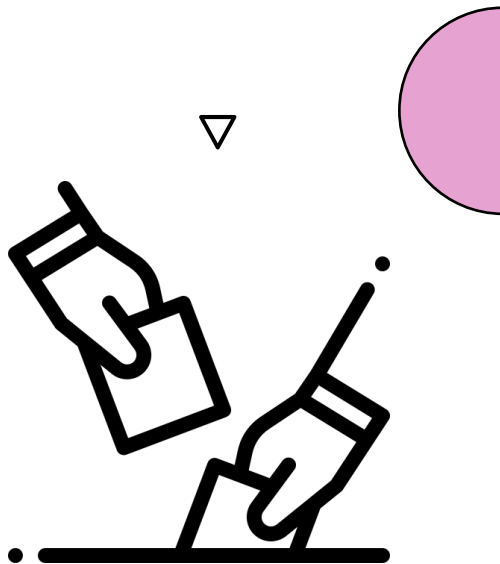
04

## La piattaforma

Immagini che mostrano il funzionamento del sistema

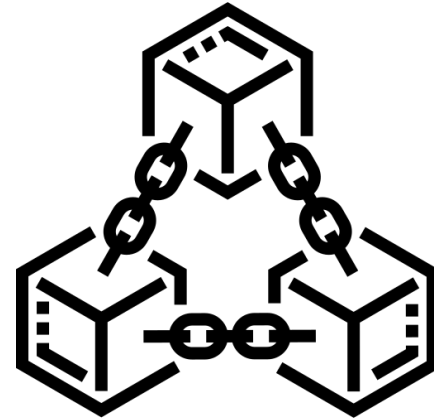
# Il problema

Le elezioni sono le fondamenta dell'espressione democratica di un popolo libero. Al giorno d'oggi, nonostante l'enorme progresso tecnologico, il processo si effettua ancora per via cartacea. Nonostante tutte le garanzie di sicurezza, il tutto è gestito dallo Stato e per questo motivo vi sono state molte controversie e più volte è stata messa in dubbio la veridicità delle elezioni stesse.



# L'idea

L'idea è stata quindi quella di sfruttare le tecnologie esistenti per creare un meccanismo sicuro di votazione che non fosse gestito da alcuna autorità (decentralizzato) in grado di garantire integrità e autenticità al voto espresso. La tecnologia che è in grado di garantire queste proprietà è la Blockchain.





01

# Cos'è?

Breve descrizione e le tecnologie  
utilizzate per l'implementazione

# SecureVoting

**SecureVoting**  
Voting made Safer

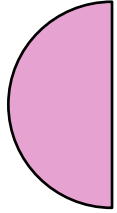


SecureVoting nasce con l'idea di  
implementare un sistema di votazioni sicuro  
basato sulla Blockchain Ethereum.

# SecureVoting

Per l'implementazione sono state usate diverse tecnologie, tra cui:

1. Ganache
2. Truffle
3. Metamask
4. Solidity
5. Web3.js





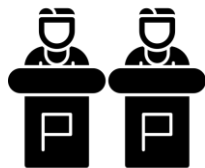


02

# Raccolta dei requisiti

Analisi delle esigenze e dei vincoli degli  
utenti

# Attori di SecureVoting



## Presidente

Colui che svolge una funzione di Amministratore, può contare i voti e aggiungere candidati



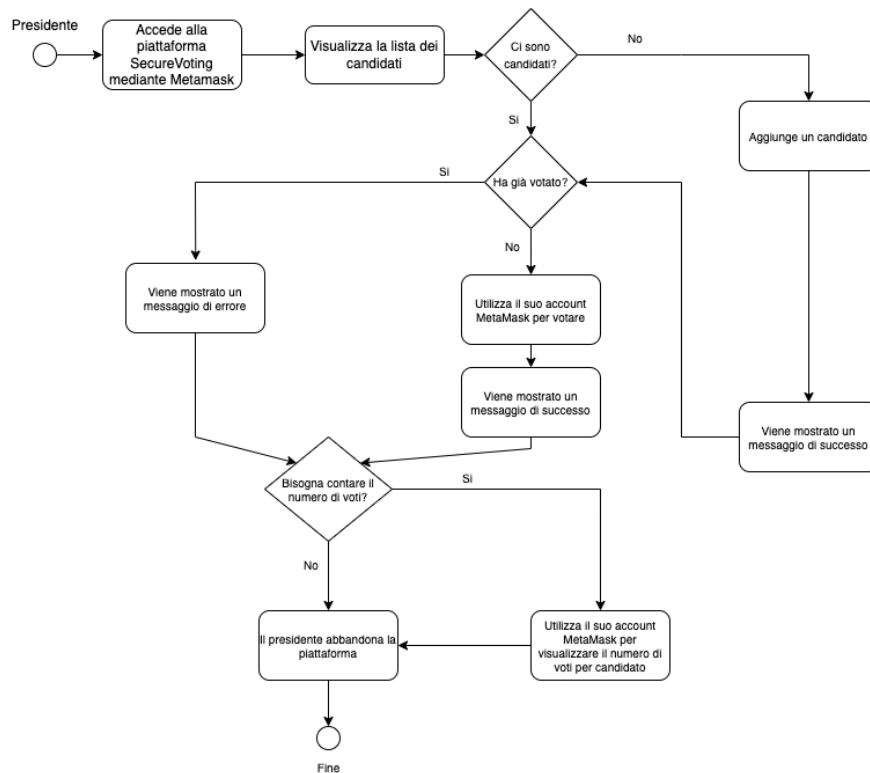
## Votante

Può effettuare una e una sola votazione in base al suo indirizzo che corrisponde ad un account sulla Blockchain

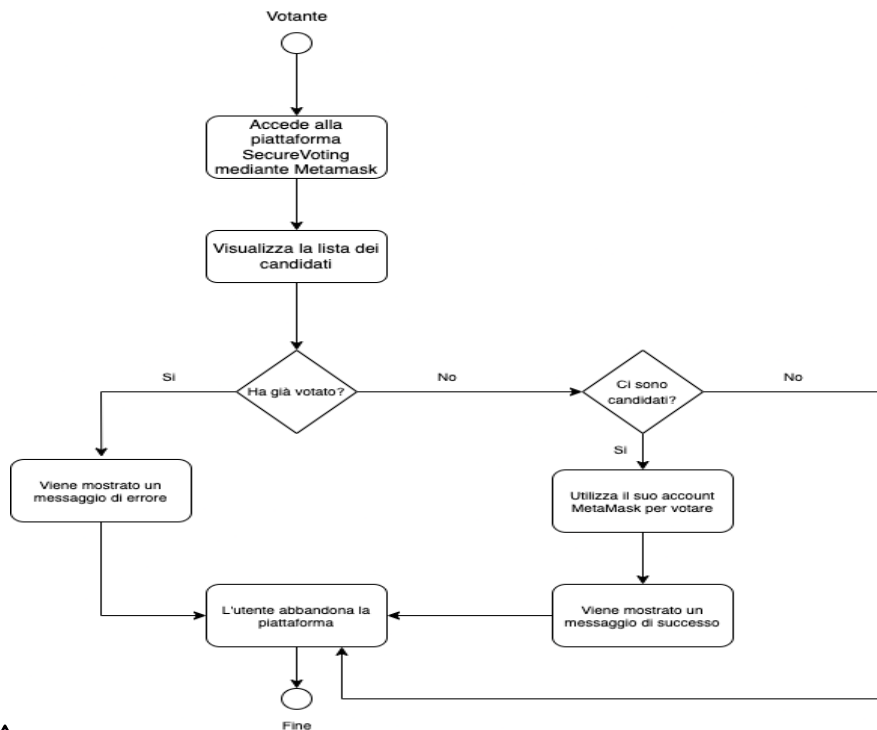
# Requisiti funzionali

ID	Requisito	Priorità
RF_1	Come presidente, è possibile aggiungere un candidato.	Alta
RF_2	Come presidente, è possibile contare il numero di voti per ogni candidato.	Alta
RF_3	Come votante, è possibile votare una ed una sola volta un candidato scelto.	Alta

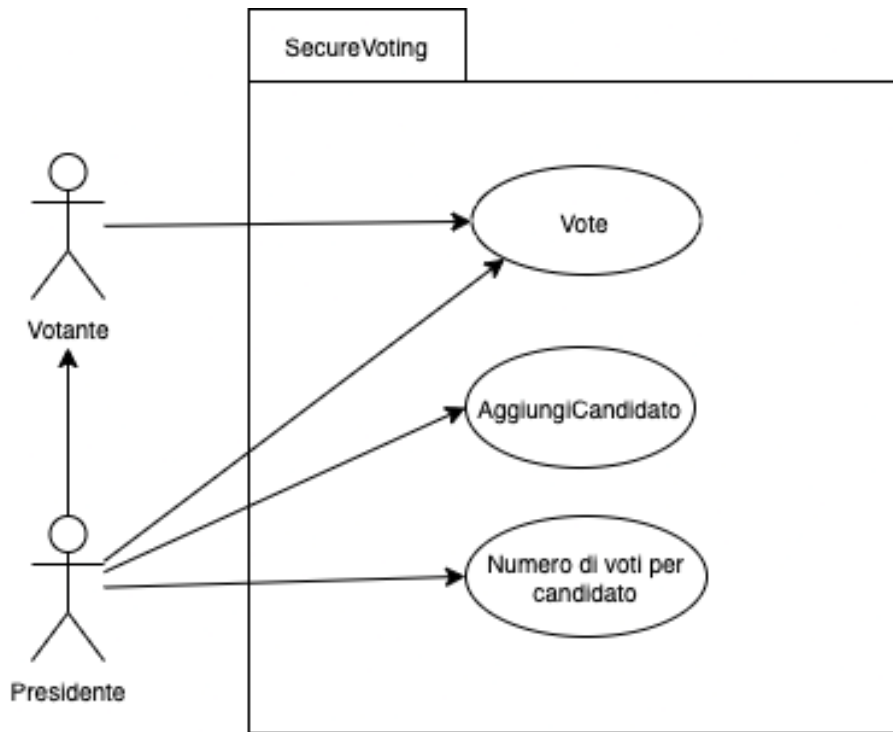
# Activity Diagram: Presidente






# Activity Diagram: Votante



# Use Case Diagram



# Use Case

Use Case (UC_1)	Sottomissione Evento 
Attore principale	Presidente
Entry Condition	Il presidente deve essere in grado di aggiungere candidati alla piattaforma.
Exit Condition On success	Il candidato viene aggiunto correttamente alla piattaforma.
Exit Condition On failure 	Il candidato non viene aggiunto alla piattaforma. 



03

# SecureVoting. sol

Implementazione dello smart contract



# 2 STRUCTs



## Candidate

- name
- party
- votes
- doesExists

uint → Candidate

## Voter



- candidateIDVote
- hasVoted

address → Voter

# isOwner()



- Controllo se l'address sender della transazione coincide con l'address che ha inizializzato il contratto. Tale informazione è salvata on chain durante la creazione del contratto

Richiamata dal FE ogni volta che viene segnalato un cambio account da Metamask, utilizzata per calcolare se il nuovo account selezionato ha i permessi di aggiungere candidati e visualizzare la count dei voti

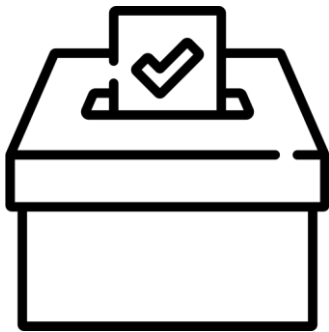
# addCandidate(name, party)



- Creo un ID per il nuovo candidato incrementando di 1 l'ultimo candidateID presente on-chain
- Salvo un nuovo Candidate con chiave il candidateID e proprietà nome, partito e numero voti inizializzato a 0



# hasAlreadyVoted()

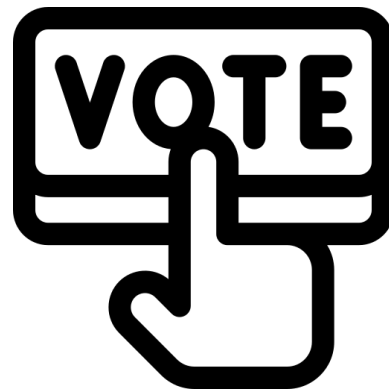


- Provo a recuperare il Voter utilizzando come chiave l'address del sender della transazione. Se il campo hasVoted è true, vuol dire che ha già votato

Utilizzata da FE prima di richiamare la funzione .vote()

# vote(candidateID)

- Incremento il numero di voti del candidato identificato tramite ID
- Salvo un nuovo Voter con chiave l'address del sender della transazione e proprietà l'ID del candidato votato

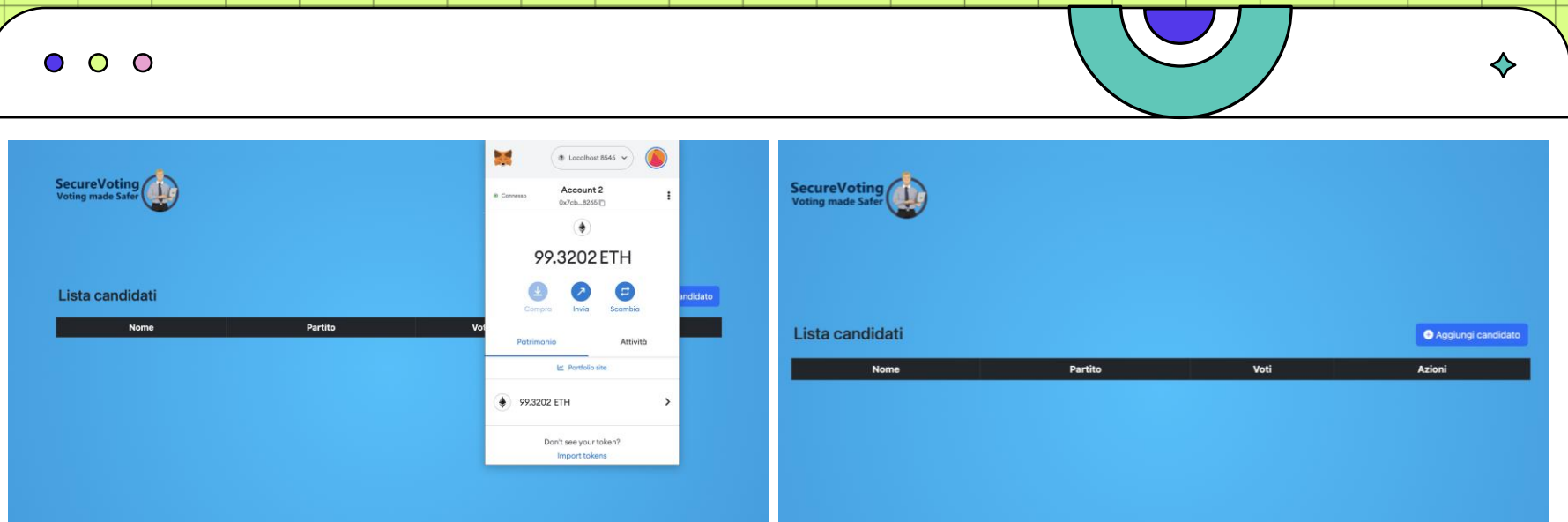




04

# La piattaforma

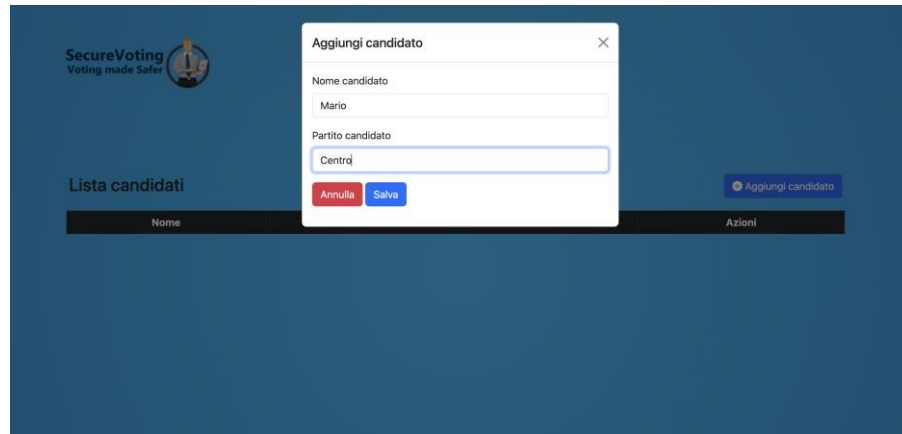
Immagini che mostrano il  
funzionamento del sistema



Account 2 è un Presidente. Da Presidente è possibile accedere al form di Aggiunta Candidato e visionare la colonna Voti per ciascun Candidato.



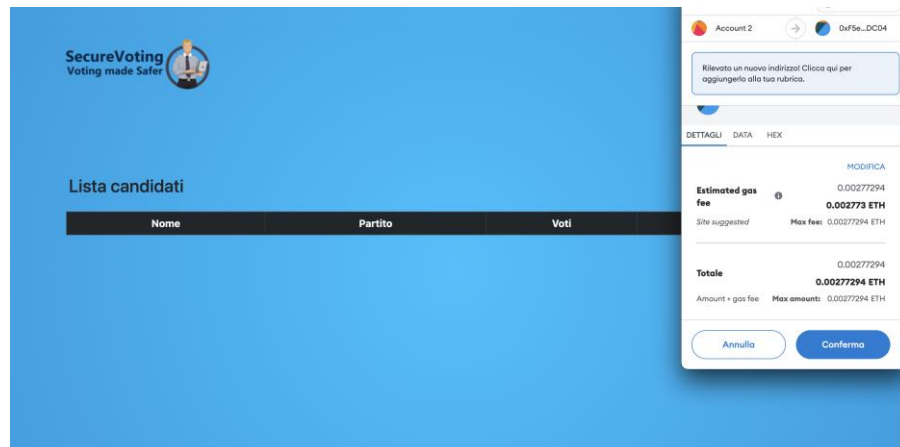
Form di aggiunta Candidato.  
È possibile inserire Nome e Partito  
del Candidato.



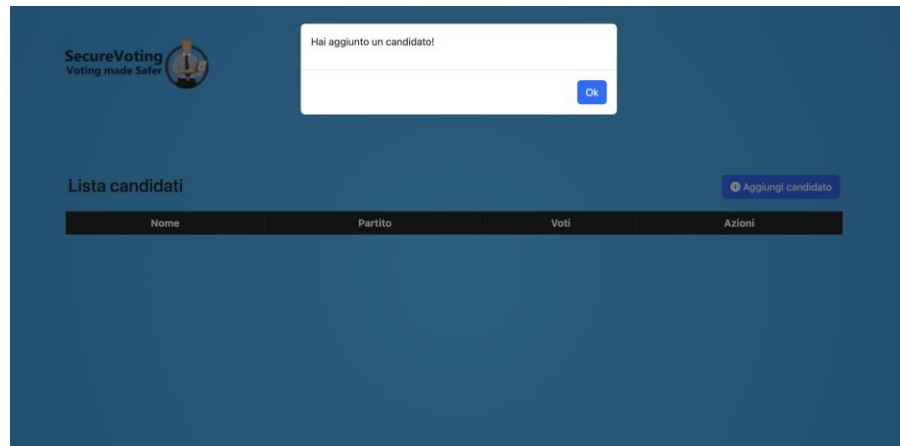
The screenshot shows a web interface for 'SecureVoting' with the tagline 'Voting made Safer'. A modal window titled 'Aggiungi candidato' is open, featuring two input fields: 'Nome candidato' (containing 'Mario') and 'Partito candidato' (containing 'Centri'). Below these fields are two buttons: a red 'Annulla' button and a blue 'Salva' button. In the background, a table titled 'Lista candidati' is partially visible, with columns for 'Nome' and 'Azioni'. A blue button labeled 'Aggiungi candidato' is also present in the background interface.



Al Salva, Metamask mostrerà una finestra di conferma con il totale delle gas fees stimate e l'account che si sta utilizzando per eseguire la transazione.




Messaggio di conferma che la  
transazione di aggiunta candidato  
si è conclusa con successo.



Da Votante posso solo osservare la lista di Candidati e scegliere chi votare.

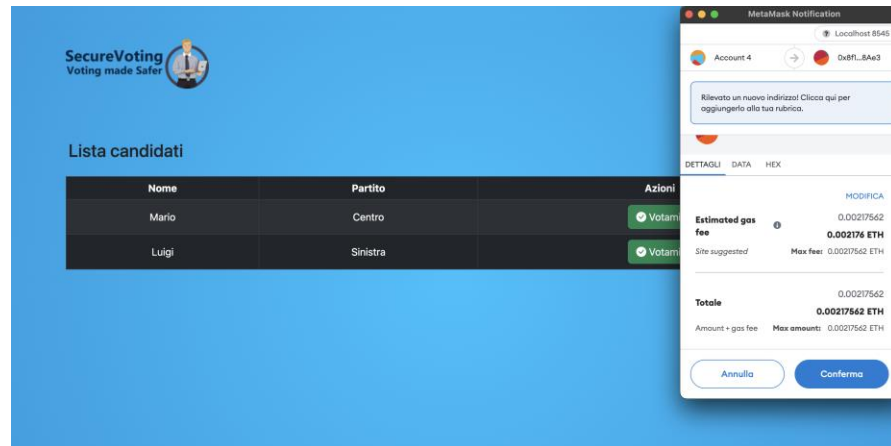
SecureVoting  
Voting made Safer

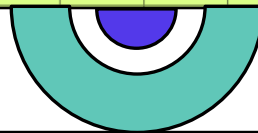


### Lista candidati

Nome	Partito	Azioni
Mario	Centro	<input checked="" type="radio"/> Votami!
Luigi	Sinistra	<input checked="" type="radio"/> Votami!


Cliccando su Votami!, appare la finestra di conferma di Metamask con le gas fees stimate per la transazione di voto.





Messaggio di conferma che la  
transazione di voto si è conclusa  
correttamente.





**SecureVoting**  
Voting made Safer

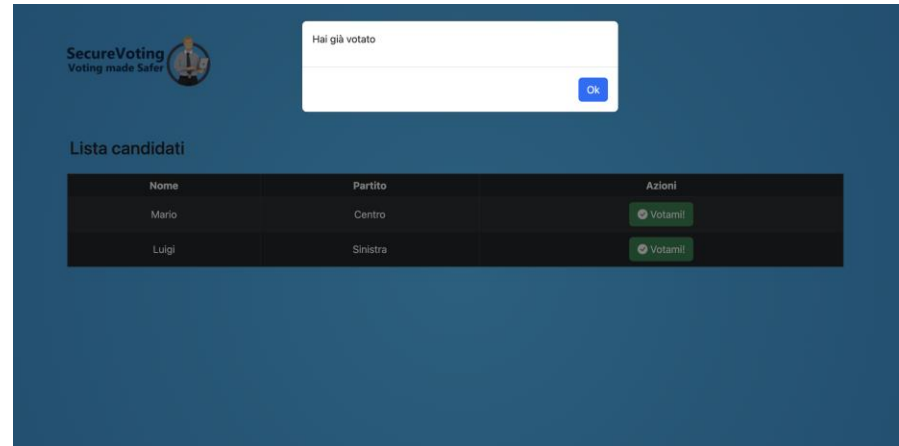
Hai votato!

Ok

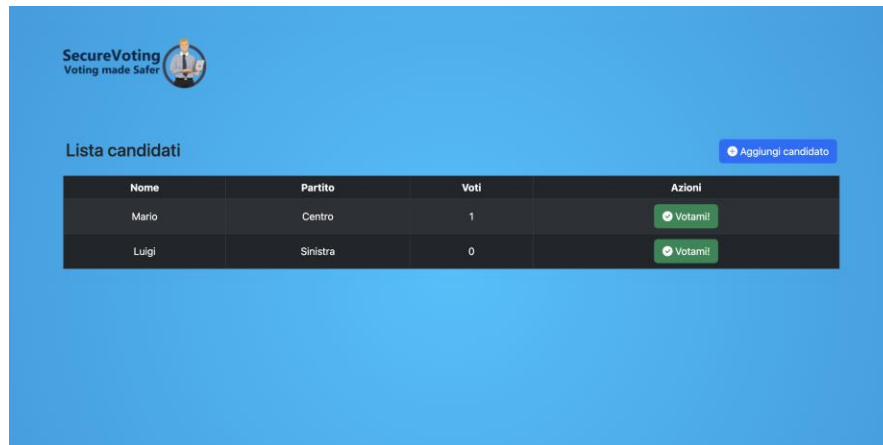
Lista candidati

Nome	Partito	Azioni
Mario	Centro	 Votami!
Luigi	Sinistra	 Votami!

Provando a votare di nuovo con lo stesso account Votante, appare un messaggio di errore che impedisce di eseguire nuovamente la transazione.



Tornando Presidente, si ha la vista con la colonna Voti ed il count aggiornato.



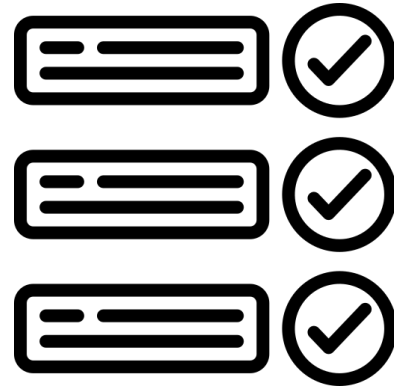
SecureVoting  
Voting made Safer

Lista candidati [+ Aggiungi candidato](#)


Nome	Partito	Voti	Azioni
Mario	Centro	1	<a href="#">Votami!</a>
Luigi	Sinistra	0	<a href="#">Votami!</a>

# Conclusioni

Il sistema da noi proposto si può applicare sia in contesti governativi che in contesti privati permettendo così di garantire autenticità al voto, salvare tempo e soldi. L'utilizzo della Blockchain garantisce quindi un sistema di votazione decentralizzato non manomettibile.







# Grazie a tutti per l'attenzione!

[g.rapa@studenti.unisa.it](mailto:g.rapa@studenti.unisa.it)

[a.cacciapuoti3@studenti.unisa.it](mailto:a.cacciapuoti3@studenti.unisa.it)

 [github.com/Givaa](https://github.com/Givaa)

 [github.com/YantCaccia](https://github.com/YantCaccia)