Voting Contract - Security Review

Contract File: https://github.com/Toyin05/Smart-Contract-Journey/blob/main/week1/Day4_VotingContract.sol

Overview:

This is a basic decentralized voting smart contract. The contract owner can add candidates. Each address can vote once. The system uses mappings, structs, and modifiers to control access and logic.

Security Features:

- Used `require()` to prevent double voting

- `onlyOwner` modifier restricts candidate addition to the deployer

- Mappings and structs improve gas efficiency and clarity

- Public view functions provide transparency with zero gas cost

Possible Improvements:

- Add a voting deadline using `block.timestamp`

- Emit events for major actions (votes cast, candidates added)

- Use enums to define election status (Open, Closed)

- Add `pause()` / circuit breaker in case of unexpected behavior

- Future-proof by making the contract upgradable (OpenZeppelin proxy pattern)

Conclusion:

The contract has a solid beginner-friendly structure. These enhancements would make it more robust and production-ready.