The brief explanations.

TokenSale.sol:

Inheritance: The contract inherits from the Ownable contract, indicating that the ownership functionality is used from a separate contract.

Sale Phases: The use of an enum (SalePhase) to manage different phases of the token sale (NotStarted, Presale, PublicSale, Finished).

Contribution Limits: Minimum and maximum contribution limits are implemented to manage the amount participants can contribute during the sale.

Refund Mechanism: Participants can claim a refund if certain conditions are met, such as the sale phase not being finished and having made a contribution.

TokenSwap.sol:

SafeERC20: The contract uses SafeERC20 for safe interactions with ERC-20 tokens, which helps prevent common vulnerabilities like reentrancy.

Exchange Rate: Implements a simple token swap with a specified exchange rate between two ERC-20 tokens.

Owner Controls: The contract includes functions to set the exchange rate, ensuring that only the owner has the ability to modify critical parameters.

VotingSystem.sol:

Owner and Voter Management: The contract keeps track of the owner and registered voters. Owners can add candidates and voters can cast votes.

Transaction Pattern: Implements a common pattern for registering, voting, and checking votes for candidates.

Quorum Concept: The contract uses a quorum concept, where a certain number of approvals are required for a transaction to be executed.

MultiSignatureWallet.sol:

Multi-Signature Wallet: Designed as a multi-signature wallet where transactions require approval from a specified number of owners.

Transaction Execution: Implements a mechanism to execute transactions only when the required quorum of owners has approved.

Fallback Function: Includes a fallback function to receive Ether transactions, supporting various use cases.