Token.sol

- The Token contract is inherited from the ERC20 and Ownable contract of @openzeppelin.
- Total of 55000000000 tokens are minted and they are then transferred to 6 different addresses as per the specification.

Sale.sol

- Public Variables -
  - tokenAddress - the address of the TOKEN contract.
  - tokenContract - the instance of the TOKEN contract.
  - saleWallet - address of the wallet that is holding all the tokens for sale.
  - saleStarted - a bool value that tracks if the sale has started.
  - isSaleOn - another bool that tracks if the sale is paused or not.
  - saleEnded - bool that tracks if the sale has ended or not.
  - minInvestment - stores the minimum investment value (in USD) by any investor.
  - startTime - stores the start time of the sale.
  - endTime - stores the end time of the sale.
  - totalWeiraised - stores the total funds raised in the sale in wei.
  - whiteList - a mapping to keep track of whitelisted addresses.
  - Investments - a mapping that stores the total investment of all the addresses.

- To get the current exchange rates of ETH and USD, I used the Ethereum Price Feeds by chain.link. To consume price data, the smart contract referenced AggregatorV3Interface on the Kovan testnet, which defines the external functions implemented by Price Feeds.
- The owner has to call the `initialize()` , this checks the allowance of the contract from the sale wallet, if less than the balance, it'll throw an error. It then updates the variables, startTime, endTime(60 days), saleStarted, isSaleOn etc.
- Owner can pause or resume the sale pauseSale() or resumeSale().
- Owner can whitelist a list of addresses at once or a single address. Owner can also blacklist any address.
- buyToken() allows any investor to directly buy the tokens. It checks for the sale status and sender details. Checks for the balance of the saleWallet. Gets the current rate of ETH and then calculates the amount of tokens to be sent to the investors. It then adds the bonus tokens to it, based on the time. If the total tokens calculated is less than the wallet balance, it then exchanges the tokens with the incoming ETH. If the total tokens calculated is greater than the balance of the wallet, it then calculates the amount of ETH to be refunded to the investors.
  - To have got the whole balance of the wallet, the investor should have bought saleBalance*100/(100+bonusPercentage) amount of tokens.
  - Hence, the remainingTokens = ((actual tokens to have sent) - saleBalance)*100/(100 + bonusPercentage).
  - Finally the value of the remaining tokens in ETH is refunded to the investor.

  The function then increments the variables like the totalWeiRaised and Investments.

- Once the time Period of the sale is completed, the owner can call the endSale() function to end the sale.