Trust Recruit Smart contract functionality.

**TrustRecruit**

**1) NftResume.sol -** this contract is basically to create User’s Resume as NFT (ERC721 Token) on Ethereum Blockchain.

1.1) Mint: It mints new NFT Token when Params are supplied to it.

Name, Hash (resume doc hash stored on IPFS), Category, Sub Category, Price.

1.2) setResumePrice: It sets the resume price after token has minted based on the NFT Token Id.

1.3) setResumeName: It sets the resume Name after token has minted based on the NFT Token Id.

1.4) setResumeHash: It sets the resume doc’s Hash after token has minted based on the NFT Token Id.

1.5) setResumeCat: It sets the resume category and sub-category after token has minted based on the NFT Token Id.

1.6) getResume: It returns the minted NFT of a particular resume when we query the smart contract.

**2) ERC721VaultFactory.sol:** This is the Factory class to generate new Tokens for the NFTResume which wants it to be fractionalize.

**3) ResumeTokenVault.sol:** This is the main Token Vault which mints new ERC20 Tokens linked to the NFTResume. It generates new tokens every time a Resume NFT is fractionalize. The token is then transferreed to the user who buys the fractional part of the NFTResume Token. It also has methods to set auction for the NFT resume token.

A new smart contract is initialized everytime a new ERC20 token is generated through factory methods. This works as a vault for ERC20 tokens which is used in fractionalization of Resume NFT. So for each Resume NFT, a new Token vault is created. It has functions to set reserve price of NFT token, Set bid, Start Bid, end the bidding process, cash the token back to eth, transfer the token to the succesfull bidder, Claim the fees etc.

**4) Settings.sol:** It is basically used set minAuctionLength, maxAuctionLength, governance fee, curator fee, feeReceiver address etc for the trust recruit platform.