Documentation on charity contract

`donateWithChoice`

```
function donateWithChoice(uint256 _amount, uint256 _proposalId, address _tokenChoice) publ
i
```

- Allows donors to donate to a specific proposal using their choice of ERC20 token

Parameters

- `_amount`: The amount to donate.
- `_proposalId`: The ID of the proposal to donate to.
- `_tokenChoice`: The ERC20 token address used for the donation.

`addProposal`

```
function addProposal(string memory _proposalURL, uint256 _amountNeeded, address _donateToA
ddress, uint256 _durationInDays) public onlyOwner
```

- Allows the contract owner to add a new proposal for fundraising.

Parameters

- `_proposalURL`: URL containing proposal details.
- `_amountNeeded`: The fundraising goal for the proposal.
- `_donateToAddress`: The address where donations will be sent.
- `_durationInDays`: Duration of the proposal in days.

`addTokenWhitelistToProposal`

```
function addTokenWhitelistToProposal(uint256 _proposalId, address _tokenContract, uint256
 _minimumTokenRequired) public onlyOwner
```

- Allows the contract owner to whitelist tokens that can be used for a specific proposal and sets a minimum donation limit for that token.

Parameters

- `_proposalId`: The ID of the proposal.
- `_tokenContract`: The ERC20 token to whitelist.
- `_minimumTokenRequired`: The minimum donation amount for the whitelisted token.
    - e.g. usdc has 6 decimals = 1000 usdc is represented as 1000000000

`matchDonation`

```
function matchDonation(uint256 _proposalId, uint256 _amountToMatch, uint256 _ratioToMatch,
address _tokenChoice) public
```

- Allows organizations to match donations for a specific proposal.

Parameters

- `_proposalId`: The ID of the proposal.
- `_amountToMatch`: The amount the organization is willing to match.
    - e.g. usdc has 6 decimals = 1000 usdc is represented as 1000000000
- `_ratioToMatch`: The ratio at which the organization will match individual donations (in basis points).
    - 100.00% = 1:1 matching key in as 10000
    - 200.00% = 1:2 matching is keyed in as 20000
- `_tokenChoice`: The ERC20 token used for the matching.

`addTokenMatchingMinimum`

```
function addTokenMatchingMinimum(uint256 _proposalId, address _tokenContract, uint256 _minimumTokenRequired) public onlyOwner
```

- Allows the contract owner to set a minimum matching donation amount for a particular token in a specific proposal.

Parameters

- `_proposalId`: The ID of the proposal.
- `_tokenContract`: The ERC20 token address.
- `_minimumTokenRequired`: The minimum matching donation amount for the token.
  - e.g. usdc has 6 decimals = 1000 usdc is represented as 1000000000

`getDonorDonationOfSpecificToken(address _addressToCheck, address _tokenChoice)`

```
function getDonorDonationOfSpecificToken(
        address _addressToCheck,
        address _tokenChoice
    ) public view returns (uint256)
```

- Retrieves the total amount donated by a specific donor in a specific token.

Parameters:

- `_addressToCheck`: Address of the donor.
- `_tokenChoice`: Token address to check.

`checkDonorTokenContributionFromProposal`

```
function checkDonorTokenContributionFromProposal(
        uint256 _proposalId,
        address _donorAddress,
        address _tokenAddress
    ) public view returns (uint256)
```

- Checks how much a donor has contributed to a specific proposal in a specific token.

Parameters:

- `_proposalId`: ID of the proposal.
- `_donorAddress`: Address of the donor.
- `_tokenAddress`: Token address to check.

`checkOrganizationDonationsForProposal`

```
function checkOrganizationDonationsForProposal(
        uint256 _proposalId,
        address _organizationAddress
    ) public view returns (uint256)
```

- Checks how much an organization has donated for matching in a specific proposal.

Parameters:

- `_proposalId`: ID of the proposal.
- `_organizationAddress`: Address of the organization.