**ERC20 Task Demo**

**About Project**

**---------------------**

**What is ERC-20?**

The ERC-20 introduces a standard for Fungible Tokens, in other words, they have a property that makes each Token be exactly the same (in type and value) of another Token. For example, an ERC-20 Token acts just like the ETH, meaning that 1 Token is and will always be equal to all the other Tokens.

**BODY**

**---------**

The ERC-20 (Ethereum Request for Comments 20), proposed by Fabian Vogelsteller in November 2015, is a Token Standard that implements an API for tokens within Smart Contracts.

**Example functionalities ERC-20 provides:**

-->transfer tokens from one account to another

-->get the current token balance of an account

-->get the total supply of the token available on the network

-->approve whether an amount of token from an account can be spent by a third-party account.

If a Smart Contract implements the following methods and events it can be called an ERC-20 Token Contract and, once deployed, it will be responsible to keep track of the created tokens on Ethereum

**NPX install credential**

**--------------------------------**

npx create-react-app react-dapp

npm install ethers hardhat @nomiclabs/hardhat-waffle ethereum-waffle chai @nomiclabs/hardhat-ethers

npx hardhat

npx hardhat compile

npx hardhat run scripts/deploy.js --network ropsten

npm start

npm install @openzeppelin/contracts

npx hardhat node

npm i bootstrap

**Project Demo**

**Home Page Screenshot**

**Contract Address** : 0xc6ad4628f000af45df9e088dc3c18c3e68875ff1

**Test net** : Ropsten

**Wallet** : Mata mask

**Development Tool**: Hardhat

**Front End** :React js

**Token Name** :DABLU PRASAD

**Symbol** : DP

**Total Supply** :100000000000000000000

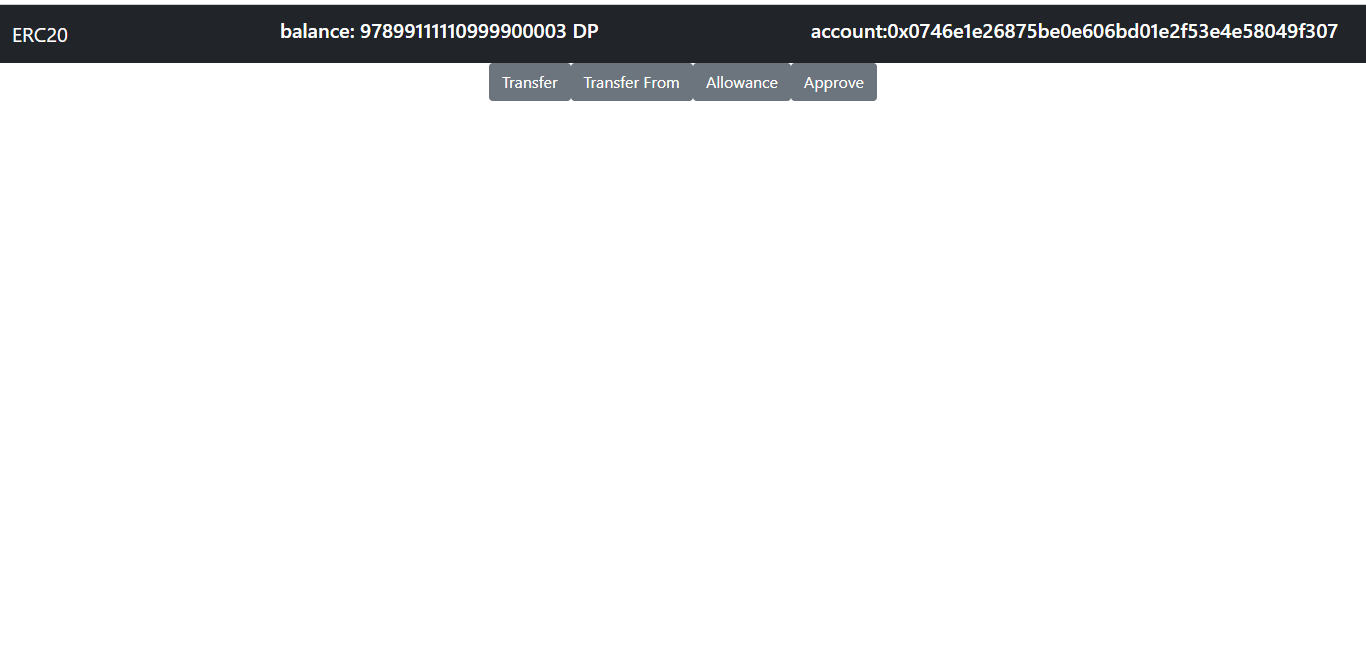
**For Interaction between frontend and smart contract :** Ether js

Home Screen

Graphical user interface, text, application

Description automatically generated **Transfer Button**

**Step1:** If you need to transfer token, then goto Transfer button and click it.



**Steps 2:** then, open another transfer component where you feed ADDRESS TO and AMOUNT you want to transfer from owner account to To account.After Feed data then click Transfer Button and Matamask transection confirm button is popup ,then confirm it.

**From Address:** Owner address (0x0746E1E26875BE0E606BD01e2F53E4E58049f307)

**To Address:** 0xf175E51Bbb18F45dd4F13F56fB16958742a14dc6

**Amount:** 1000000000000000000 (1 DP)

Graphical user interface, text, application, email

Description automatically generated

**Finally:**

Your Transaction is done and Transection Hash must be created of your transection.

**Transaction Hash:** 0xebe7e5d8a883ad86c741d678a623fbc8469982c5b1fd3761b1cb7c66f0c76697

Graphical user interface, text, application, email

Description automatically generated

**Transfer From Button**

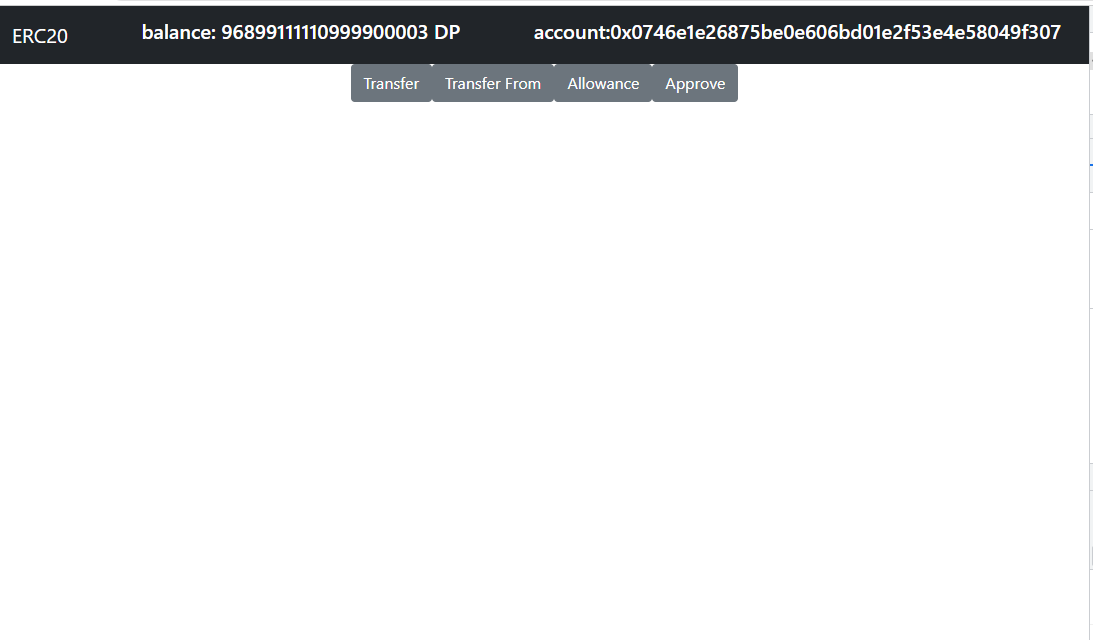
The **transferFrom** function is the peer of the **approve** function. It allows a delegate approved for withdrawal to transfer owner funds to a third-party account.

1. **Approve Button**

**approve** withdraws tokens from the owner’s address up to a certain amount of tokens.

**Steps:1**

For Approve, goto to Approve button and click it.



**Steps 2:** then,Approve component is open and feed the Spender Address and Amount you want to Approve for spend by owner Address and confirm popup of matamask is showing after click on Approve button.

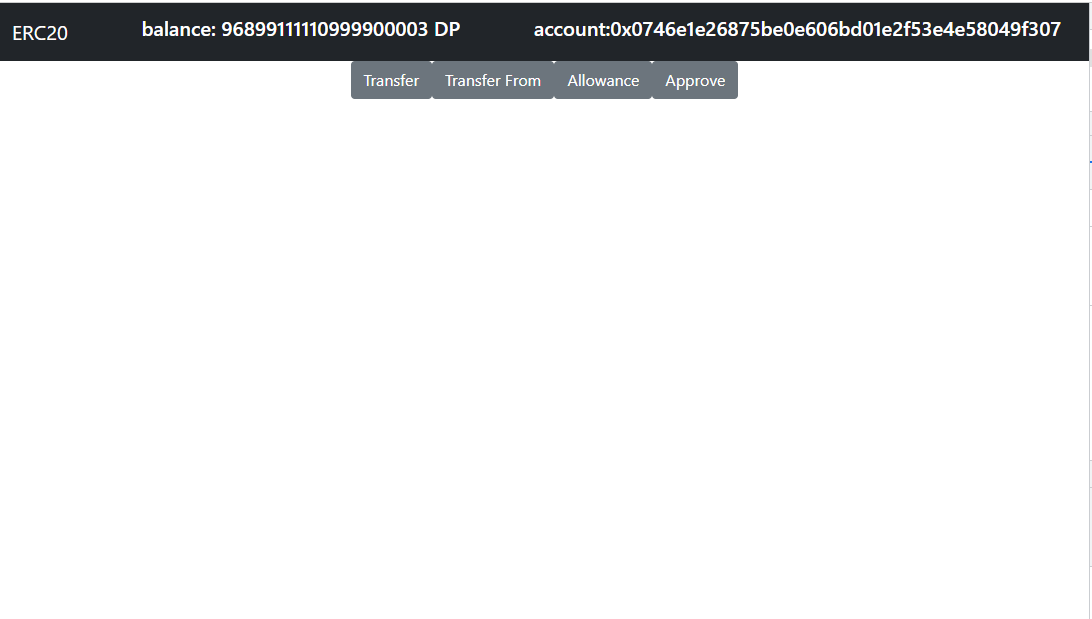
Graphical user interface, text, application, email

Description automatically generated

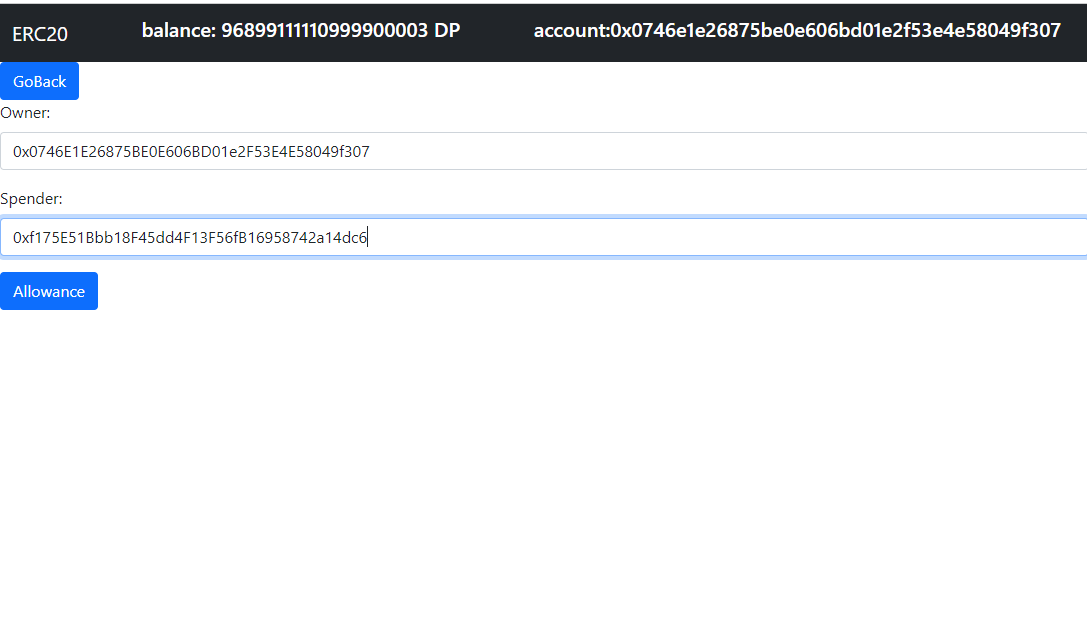
**2)Allowance**

**allowance** returns the number of tokens withdrawable from the owner’s account

**Steps1:** For Allowance, goto the Allowance button and click on it.



**Steps2:**  then, Allowance Component is open and feed Owner address and Spender Address and click on Allowance button.



**3) Transfer From**

**transferFrom** transfers a certain amount of tokens from a beneficiary address to a recipient address

**Steps1**: For Transfer From, first approve the spender token and the amount which spender spend by owner address,goto transfer from button and click on it.

Graphical user interface, text

Description automatically generated

**Steps2**: then, TransferFrom Component is open , where you feed Sender Address(Owner address) and Reciptent Address and also Amount which is you want to transfer and confirm the matamask popup after clicking Transfer from button.

Graphical user interface, text, application, email

Description automatically generated

**Finally :** We get Transection hash of given transection perform on Ropsten Testnet of TransferFrom Button.

**Transection hash** : 0xf730a9c2689cd0920dfa9debc55cb623fdd3d654f726eda8d38e53f3d9c146f9

Graphical user interface, text, application, email

Description automatically generated

Thanks You