

Blockchain Project

Exchange Calculator

Chainlink Fall Hackathon 2021

<https://devpost.com/software/exchange-calculator>

Exchange Calculator



Exchange Calculator, Tool for development

1. Hardhat for solidity
2. Deploy to alchemy(KOVAN)
3. Frontend(Ether.js) used reactjs
4. Chainlink Ethereum Data Feeds

<https://www.youtube.com/watch?v=tRGKI0S7gSU>

Exchange Calculator

Exchange Calculator

Connect wallet

0x597...8F7

ETH/LINK:

1 ETH

ETH/LINK

ETH/BTC:

100

ETH/BTC

ETH/DAI:

1 ETH

ETH/DAI

ETH/USDT:

1 ETH

ETH/USDT

ETH/BUSD:

1 ETH

ETH/BUSD

7.5096680632539194 BTC

ETH/BTC, Feed data from Ethereum Data Feeds

Exchange Calculator

DEVPOST

Hackathons

Projects

Search hackathons

Host a hackathon

Log in

Sign up

Exchange Calculator


Exchange Calculator ETH to another coin

Like

Comment

STORY

UPDATES

 Weera Hackathon 2021 Project

Exchange Calculator

Connect wallet
0x597...8F7

Watch later

Share

ETH/LINK:
1 ETH
ETH/LINK

ETH/BTC:
1 ETH
ETH/BTC

ETH/DAI:
1 ETH
ETH/DAI


ETH/USDT:
1 ETH
ETH/USDT

ETH/BUSD:
1 ETH
ETH/BUSD


0

Watch on YouTube

SUBMITTED TO

 Chainlink Fall Hackathon 2021

CREATED BY

 weera saebe

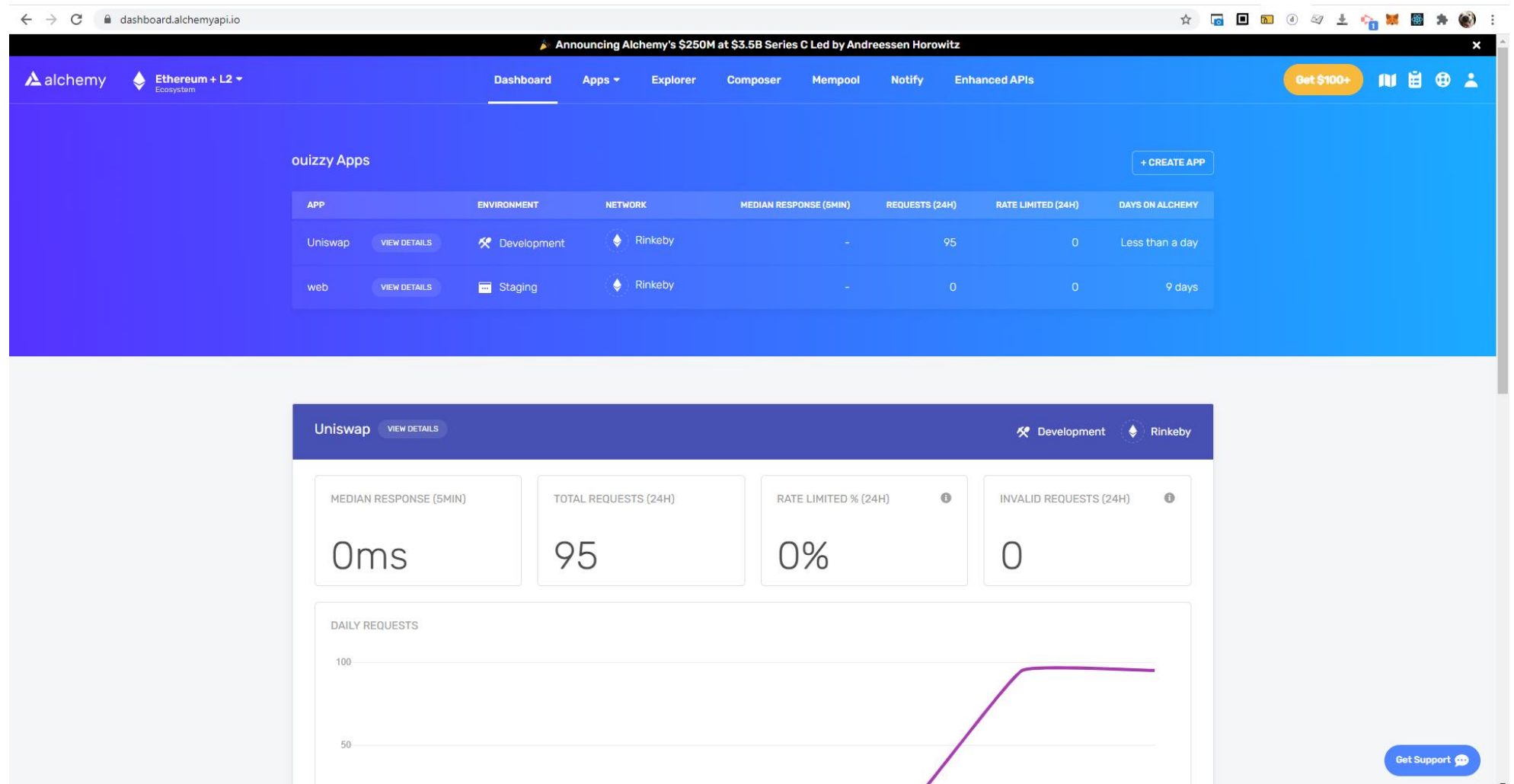
More details, Please visit at the Chainlink Fall Hackathon :

<https://devpost.com/software/exchange-calculator>

Blockchain Project

Vote

Vote Project



Vote system, Tool for development

1. Hardhat for develop solidity
2. Alchemy for deploy this project (Rinkeby test network)
3. React for develop frontend

Vote Project

The image shows a development environment with three main components:

- Terminal (Left):** Displays the output of Hardhat commands. The first part shows a successful deployment to the Rinkeby test network. The second part shows a failed deployment attempt due to a Solidity version mismatch (0.8.7 vs 0.8.10). The third part shows a successful deployment to the Rinkeby test network.
- Dashboard (Middle):** A dashboard for the 'Vote Project' showing success and failure rates. The success rate is 100% (LAST 1 H) and 28 (TOTAL REQUESTS LAST 24 H). The failure rate is 0% (LAST 1 H) and 0 (INVALID REQUESTS LAST 24 H).
- Wallet (Right):** A wallet interface for the Rinkeby Test Network. It shows the account name 'Account 1' with address '0x5977a3af2e8289fd685289c1cb2321cbd779b8f7'. The balance is 13.2644 ETH. There are buttons for Buy, Send, and Swap. Below the balance, there are transactions: 'Add Liquidity E T H' and 'Approve DAI spend li...'. A red box highlights the account name and address, and a red arrow points from this box to the 'From' field in the transaction details.

The transaction details for the first transaction (eth_getTransactionByHash) are as follows:

PARAMS	RAW REQUEST	RESULT	RAW RESPONSE
0: 0x4aae0986e8679fda28c2c2e0f645c2d05a6476ff4a25915874dd009d864f76d		Block Hash: null	
		Block Number: null	
		From: 0x5977a3af2e8289fd685289c1cb2321cbd779b8f7	
		Gas: 12000000	
		Gas Price: 1000000000	
		Hash: 0x4aae0986e8679fda28c2c2e0f645c2d05a6476ff4a25915874dd009d864f76d	
		Input: 0x60806040523480156200001157600000fd5b50604051620017d138...	
		Max Fee Per Gas: 1000000000	
		Max Priority Fee Per Gas: 1000000000	

Deploy application to Rinkeby test network via Alchemy and input rinkeby url , private key for deployment

Vote Project

The screenshot displays the Etherscan interface for a transaction on the Rinkeby Testnet. The transaction hash is 0x4aae0986e8679fda28c2c2e0f645c2d05a6476ff4a25915874dd0009d864f76d. The status is 'Success' with 33 block confirmations. The transaction was created 8 minutes ago. The 'To' field shows a contract address: [Contract 0x725a9dc5d95d3f233e84536b6a05bb5224dd9b64 Created]. The value is 0 Ether (\$0.00), the transaction fee is 0.001280675014087425 Ether (\$0.00), and the gas price is 0.000000001000000011 Ether (1.000000011 Gwei).

Overlaid on the right is a terminal window titled 'MINGW64/d/blockchian_ballot/ballot'. It shows the execution of the following commands and their outputs:

```
QUY_98LAPTOP-80D8HCHN MINGW64 /d/blockchian_ballot/ballot (main)
$ npx hardhat run scripts/deploy-ballot3.ts --network rinkeby
Generating typings for: 0 artifacts in dir: src/types for target: ethers-v5
Successfully generated 3 typings!
Successfully generated 3 typings for external artifacts!
Ballot deployed to: 0xa9a38a6c7839198c31bc01f46e24bf687f87

QUY_98LAPTOP-80D8HCHN MINGW64 /d/blockchian_ballot/ballot (main)
$ npx hardhat compile
Solidity 0.8.7 is not fully supported yet. You can still use Hardhat, but some features, like stac
ght not work correctly.

Learn more at https://hardhat.org/reference/solidity-support"

Compiling 1 file with 0.8.7
Generating typings for: 1 artifacts in dir: src/types for target: ethers-v5
Successfully generated 5 typings!
Successfully generated 3 typings for external artifacts!
Compilation finished successfully

QUY_98LAPTOP-80D8HCHN MINGW64 /d/blockchian_ballot/ballot (main)
$ npx hardhat run scripts/deploy-ballot3.ts --network rinkeby
Generating typings for: 0 artifacts in dir: src/types for target: ethers-v5
Successfully generated 3 typings!
Successfully generated 3 typings for external artifacts!
Ballot deployed to: 0x725A9DC5d95d3f233e84536b6a05BB5224dd9b64

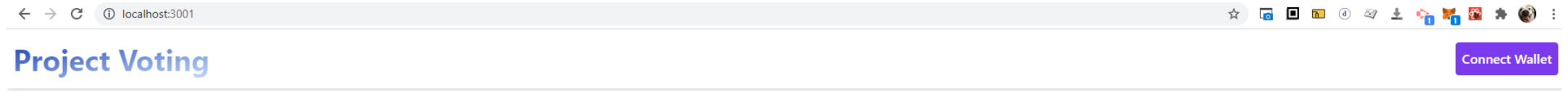
QUY_98LAPTOP-80D8HCHN MINGW64 /d/blockchian_ballot/ballot (main)
$
```

The line 'Ballot deployed to: 0x725A9DC5d95d3f233e84536b6a05BB5224dd9b64' in the terminal is highlighted with a red box.

At the bottom of the Etherscan page, there is a footer with the text 'Powered by Ethereum', a cookie notice, and links for 'Preferences' and 'Get It'.

You can check the deployment details at <https://etherscan.io/> and insert the hash value from deploy

Vote Project

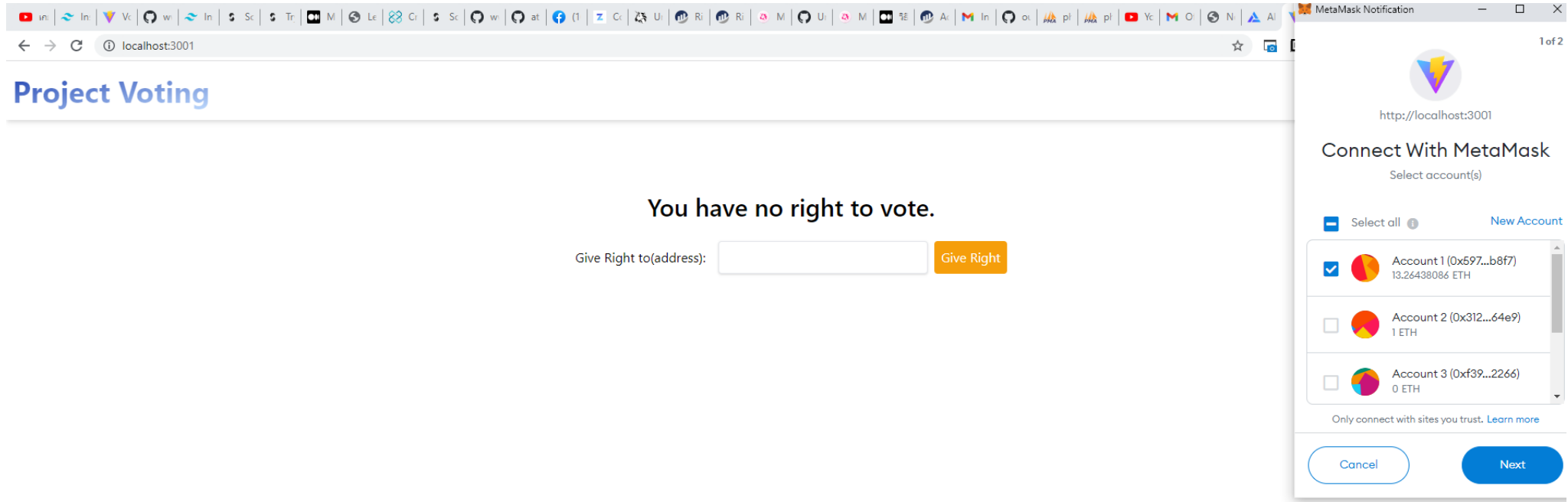


When deploy is completed, open the frontend.

Conditional for vote system:

Admin (who deploys the smart contract is admin) is required to give each person the right to vote, otherwise you cannot vote.


Vote Project





First of all you have to connect your Metamask.

Vote Project

Project Voting 0x597...8F7

Nodejs

Count: 0 Vote

Golang

Count: 0 Vote

Solidity

Count: 0 Vote

You have 1 rights.


Give Right to(address): Give Right

When you connect metamask, you can vote.

Vote Project


Project Voting

Nodejs




Count: 0 Vote

Golang



Count: 0 Vote

Solidity



Count: 0 Vote

You have 1 rights.


Give Right to(address):

Give Right

MetaMask Notification

Rinkeby Test Network

Account 1



0x725...9b64

New address detected! Click here to add to your address book.

http://localhost:3001

VOTE

DETAILS

DATA

Estimated gas fee

0.0001428 0.0001428 ETH

Site suggested

Likely in < 30 seconds

Max fee:

0.0001428 ETH

Total

0.0001428 0.0001428 ETH

Amount + gas fee

Max amount: 0.0001428 ETH

Reject

Confirm


When voting connects a metamask that shows the details of the gas.
Because when write data into the blockchain it have to gas costs.

Vote Project

Project Voting

0x597...8F7


Nodejs



Count: 0

Vote


Golang



Count: 0

Vote

Solidity



Count: 1

Voted

You have voted to Solidity.

Give Right to(address):

Give Right

Voted

Vote Project

The screenshot displays a web browser window at localhost:3001. The main content area is titled "Project Voting" and features three voting cards:

- Nodejs**: Count: 0, Vote button (disabled).
- Golang**: Count: 0, Vote button (disabled).
- Solidity**: Count: 1, Voted button (active).

Below the cards, a confirmation message reads: "You have voted to Solidity." Underneath this message is a form labeled "Give Right to(address):" with an input field containing the address "0x312A09E095Ff9caD42035Da4E" and a "Give Right" button.

On the right side of the browser window, a MetaMask wallet interface is open for "Account 2" (0x312...64e9). It shows a balance of 1 ETH and options to Buy, Send, or Swap. The transaction history section indicates "You have no transactions." At the bottom, there is a link to "Need help? Contact MetaMask Support".


If the 2nd account is voting, it can't vote. It needs to be granted first from admin.

Admin bring the address(2nd) to input box and submit.

Vote Project


Project Voting

Nodejs




Count: 0 Vote

Golang



Count: 0 Vote

Solidity



Count: 1 Voted

You have voted to Solidity.

Give Right to(address):

0x312A09E095Ff9caD42035Da4E

Give Right

MetaMask Notification

Rinkeby Test Network

Account 1

0x725...9b64

New address detected! Click here to add to your address book.

http://localhost:3001

GIVE RIGHT TO VOTE

DETAILS

DATA

Estimated gas

0.00007488

0.00007488 ETH

fee

Site suggested

Likely in <30 seconds

Max fee:

0.00007488 ETH

Total

0.00007488

0.00007488 ETH

Amount + gas fee

Max amount: 0.00007488 ETH

Reject


Confirm

When account admin give right, the 2nd account will show details of gas.

Vote Project

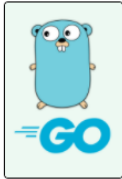
Project Voting

Nodejs




Count: 0 Vote

Golang



Count: 0 Vote

Solidity



Count: 1 Vote

You have 1 rights.

Give Right to(address):

0x312A09E095Ff9caD42035Da4E

Give Right

MetaMask Notification

Rinkeby Test Network

Account 2

0x725...9b64

New address detected! Click here to add to your address book.

http://localhost:3001

VOTE

DETAILS

DATA

Estimated gas fee

0.00011715

0.00011715 ETH

Site suggested

Likely in < 30 seconds

Max fee:

0.00011715 ETH

Total

0.00011715

0.00011715 ETH

Amount + gas fee

Max amount: 0.00011715 ETH

Reject


Confirm

Switching to a second account, you can now vote and vote.

Vote Project


Project Voting

Nodejs





Count: 0 Vote

Golang



Count: 0 Vote

Solidity 



Count: 2 Voted


You have voted to Solidity.

Give Right to(address):

0x312A09E095Ff9caD42035Da4E


Give Right

Vote

 to:0x312A09E095Ff9caD42035Da4BC

voted for Solidity | total score: 2

Vote

 to:0x312A09E095Ff9caD42035Da4BC

voted for Solidity | total score: 2

Result voting

Vote Project

The screenshot shows the Etherscan website interface for the Rinkeby Testnet Network. The browser's address bar displays the URL: `rinkeby.etherscan.io/tx/0x9e711c9ce0406b28c2016471d8bf13d52a348e3794db8c4e521ad2c435b8b8c7`. The page title is "Transaction Details". The transaction hash is `0x9e711c9ce0406b28c2016471d8bf13d52a348e3794db8c4e521ad2c435b8b8c7`. The status is "Success". The block number is `9696427` with "2 Block Confirmations". The timestamp is "52 secs ago (Nov-24-2021 04:54:40 AM +UTC)". The transaction is from address `0x312a09e095ff9cad42035da4bc6c4475dac464e9` to contract `0x725a9dc5d95d3f233e84536b6a05bb5224dd9b64`. The value is `0 Ether ($0.00)`, the transaction fee is `0.0001515139999219 Ether ($0.00)`, and the gas price is `0.000000001939999999 Ether (1.939999999 Gwei)`. A note at the bottom states: "A transaction is a cryptographically signed instruction from an account that changes the state of the blockchain. Block explorers track the details of all transactions in the network. Learn more about transactions in our Knowledge Base."

Etherscan

Rinkeby Testnet Network

All Filters Search by Address / Txn Hash / Block / Token / Ens

Home Blockchain Tokens Misc Rinkeby

Transaction Details

Overview Logs (1) Access List State

[This is a Rinkeby **Testnet** transaction only]

Transaction Hash:	0x9e711c9ce0406b28c2016471d8bf13d52a348e3794db8c4e521ad2c435b8b8c7
Status:	Success
Block:	9696427 2 Block Confirmations
Timestamp:	52 secs ago (Nov-24-2021 04:54:40 AM +UTC)
From:	0x312a09e095ff9cad42035da4bc6c4475dac464e9
To:	Contract 0x725a9dc5d95d3f233e84536b6a05bb5224dd9b64
Value:	0 Ether (\$0.00)
Transaction Fee:	0.0001515139999219 Ether (\$0.00)
Gas Price:	0.000000001939999999 Ether (1.939999999 Gwei)

[Click to see More](#)

A transaction is a cryptographically signed instruction from an account that changes the state of the blockchain. Block explorers track the details of all transactions in the network. Learn more about transactions in our [Knowledge Base](#).

When doing various transactions, the hash value is obtained. The hash value can be checked at etherscan.

Blockchain Project Bank

Bank Project

Address : 0x

Status:

Connect Wallet

I'm Bank

Deposit Amount:

Amount

Deposit

Withdraw Amount:

Amount

Deposit

Checkbalance:

Your Balance

MINGW64/d/hackton

```
GUI_90LAPTOP-80D8HCHN MINGW64 /d/blockchian_ballot
$ cd ../
GUI_90LAPTOP-80D8HCHN MINGW64 /d
$ cd hackton/
GUI_90LAPTOP-80D8HCHN MINGW64 /d/hackton (main)
$ npx hardhat run scripts/sample-script.ts --network localhost
Generating typings for: 0 artifacts in dir: src/types for target: ethers-v5
Successfully generated 3 typings!
HardhatError: HH108: Cannot connect to the network localhost.
Please make sure your node is running, and check your internet connection and network
at HttpProvider._fetchJsonRpcResponse (D:\hackton\node_modules\hardhat\src\internal\
ders\http.ts:176:15)
    at processTicksAndRejections (node:internal/process/task_queues:96:5)
    at HttpProvider.request (D:\hackton\node_modules\hardhat\src\internal\core\prov
5:29)
    at GanacheGasMultiplierProvider._isGanache (D:\hackton\node_modules\hardhat\src
providers\gas-providers.ts:315:30)
    at GanacheGasMultiplierProvider.request (D:\hackton\node_modules\hardhat\src\in
viders\gas-providers.ts:304:23)
    at ChainIdValidatorProvider._getChainIdFromEthNetVersion (D:\hackton\node_modul
internal\core\providers\chainId.ts:33:17)
    at ChainIdValidatorProvider._getChainId (D:\hackton\node_modules\hardhat\src\in
viders\chainId.ts:17:25)
    at ChainIdValidatorProvider.request (D:\hackton\node_modules\hardhat\src\intern
rs\chainId.ts:55:29)
    at EthersProviderWrapper.send (D:\hackton\node_modules\@nomiclabs\hardhat-ether
ethers-provider-wrapper.ts:13:20)
    at Object.getSigners (D:\hackton\node_modules\@nomiclabs\hardhat-ethers\src\int
s:23:20)
    Caused by: FetchError: request to http://localhost:8545/ failed, reason: connec
127.0.0.1:8545
    at ClientRequest.<anonymous> (D:\hackton\node_modules\node-fetch\lib\index.
    at ClientRequest.emit (node:events:390:28)
    at ClientRequest.emit (node:domain:475:12)
    at Socket.socketErrorListener (node:_http_client:447:9)
    at Socket.emit (node:events:390:28)
    at Socket.emit (node:domain:475:12)
    at emitErrorNT (node:internal/streams/destroy:157:8)
    at emitErrorCloseNT (node:internal/streams/destroy:122:3)
    at processTicksAndRejections (node:internal/process/task_queues:83:21)
GUI_90LAPTOP-80D8HCHN MINGW64 /d/hackton (main)
$ npx hardhat run scripts/sample-script.ts --network localhost
Generating typings for: 0 artifacts in dir: src/types for target: ethers-v5
Successfully generated 3 typings!
Successfully generated 3 typings for external artifacts!
Foo: Foo
Greeter deployed to: 0x5FbDB2315678afecb367F032d93F642F64180aa3
GUI_90LAPTOP-80D8HCHN MINGW64 /d/hackton (main)
$ |
```

C:\Windows\system32\cmd.exe

```
GUI_90LAPTOP-80D8HCHN MINGW64 /d/hackton (main)
$ npx hardhat node
Started HTTP and WebSocket JSON-RPC server at http://127.0.0.1:8545/
Accounts
=====
WARNING: These accounts, and their private keys, are publicly known.
Any funds sent to them on Mainnet or any other live network WILL BE LOST.

Account #0: 0xf39fd6e51aad88f64ce6ab8827279cffffb92266 (10000 ETH)
Private Key: 0xac0974bec39a17e36ba4a6b4d238ff94bacc478cded5efcae774d7bf4f2ff80

Account #1: 0x70997970c51812dc3a010c7d01b50e0d17dc79c8 (10000 ETH)
Private Key: 0x59c6995e989f97a5a0044966f0945389dc9e86dae88c7a8412f4603b6b78690d

Account #2: 0x3c44cdddb6a900fa2b585dd299e03d12fa4293bc (10000 ETH)
Private Key: 0x5de4111afa1a4b94908f83103eb1f1706367c2e68ca870fc3fb9a804cdab365a

Account #3: 0x90f79bfe6b2c4f870365e785982e1f101e93b906 (10000 ETH)
Private Key: 0x7c852118294e51e653712a81e05800f419141751be58f605c371e15141b007a6

Account #4: 0x15d344fa54267db7d7c367839aaf71a00a2c6a65 (10000 ETH)
Private Key: 0x47e179ec197488593b187f80a00eb0da91f1b9d0b13f8733639f19c30a34926a

Account #5: 0x9965507d1a55bcc2695c58ba16fb37d819b0a4dc (10000 ETH)
Private Key: 0x8b3a350cf5c34c9194ca85829a2df0ec3153be0318b5e2d3348e872092edffba

Account #6: 0x976ea74026e726554db657fa54763abd0c3a0aa9 (10000 ETH)
Private Key: 0x92db14e403b83dfe3df233f83dfa3a0d70996f21ca9b0d66b8d88b2b4ec1564e

Account #7: 0x14dc79964da2c08b23698b3d3cc7ca32193d9955 (10000 ETH)
Private Key: 0x4bbbf85ce3377467afe5d46f804f221813b2bb87f24d81f601fcd9b7fcbf4356

Account #8: 0x23618e81e3f5cdf7f54c3d65f7fbc0abf5b21e8f (10000 ETH)
Private Key: 0xd8bda1821b80551c9d65939329250298aa3472ba22f6e921c0cf5d620ea7b97

Account #9: 0xa0ee7a142d267c1f36714e4a8f75612f20a79720 (10000 ETH)
Private Key: 0x2a871d0798f97d79848a013d4936a73bf4cc922c825d33c1cf7073dff6d409c6

Account #10: 0xbcd4042de499d14e55001ccbb24a551f3b954096 (10000 ETH)
Private Key: 0xf214f2b2cd398c806f84e317254e0f0b801d0643303237d97a22a48e01628897

Account #11: 0x71be63f3384f5fb9895e898a86b02fb2426c5788 (10000 ETH)
Private Key: 0x701b615bbdfb9de65240bc28bd21bbc0d996645a3dd57e7b12bc2bdf6f192c82

Account #12: 0xfabb0ac9d68b0b445fb7357272ff202c5651694a (10000 ETH)
Private Key: 0xa267530f49f8280200edf313ee7af6b827f2a8bce2897751d06a843f644967b1

Account #13: 0x1cbd3b2770909d4e10f157cab84c7264073c9ec (10000 ETH)
Private Key: 0x47c99abed3324a2707c28afff71267e45918ec8c3f20b8aa892e8b065d2942dd

Account #14: 0xdf3e18d64bc6a983f673ab319ccae4f1a57c7097 (10000 ETH)
Private Key: 0xc526ee95bf44d8fc405a158bb884d9d1238d99f0612e9f33d006bb0789009aaa
```

Hardhat run

Hardhat node

Basic Bank system, Tool for development

1. Hardhat for develop solidity
2. Hardhat node for deploy this project
3. React for develop frontend

Bank Project

localhost:3001

Address : 0xf39Fd6e51aad88F6F4ce6aB8827279cFfb92266

Status:

Connect Wallet

I'm Bank

Deposite Amount:

10

Deposite

Withdraw Amount:

Amount

Deposite

Checkbalance:

Your Balance

```

OUI_9@LAPTOP-80D8HCHN MINGW64 /d/hackton (main)
$ npx hardhat node
Started HTTP and WebSocket JSON-RPC server at http://127.0.0.1:8545/

Accounts
=====

WARNING: These accounts, and their private keys, are publicly known.
Any funds sent to them on Mainnet or any other live network WILL BE LOST.

Account #0: 0xf39Fd6e51aad88F6F4ce6aB8827279cFfb92266 (10000 ETH)
Private Key: 0xac0974bec39a17e36ba4a6b4d238ff944bacb478cbed5efcae784d7bf4f2ff80

Account #1: 0x70997970c51812dc3a010c7d01b50e0d17dc79c8 (10000 ETH)
Private Key: 0x59c6995e998f97a5a0044966f0945389dc9e86dae88c7a8412f4603b6b78690d

Account #2: 0x3c44cdddb6a900fa2b585dd299e03d12fa4293bc (10000 ETH)
Private Key: 0x5de4111afa1a4b94908f83103eb1f1706367c2e68ca870fc3fb9a804cdab365a

Account #3: 0x90f79bf6eb2c4f870365e785982e1f101e93b906 (10000 ETH)
Private Key: 0x7c852118294e51e653712a81e05800f419141751be58f605c371e15141b007a6

Account #4: 0x15d34aaf54267db7d7c367839aaf71a00a2c6a65 (10000 ETH)
Private Key: 0x47e179ec197488593b187f80a00eb0da91f1b9d0b13f8733639f19c30a34926a

Account #5: 0x9965507d1a55bcc2695c58ba16fb37d819b0a4dc (10000 ETH)
Private Key: 0x8b3a350cf5c34c9194ca85829a2df0ec3153be0318b5e2d3348e872092edffba

Account #6: 0x976ea74026e726554db657fa54763abd0c3a0aa9 (10000 ETH)
Private Key: 0x92db14e403b83dfe3df233f83dfa3a0d7096f21ca9b0d6d6b8d88b2b4ec1564e

Account #7: 0x14dc79964da2c08b23698b3d3cc7ca32193d9955 (10000 ETH)
Private Key: 0x4bbbf85ce3377467afe5d46f804f221813b2bb87f24d81f60f1fcd9f7cbf4356

Account #8: 0x23618e81e3f5cdf7f54c3d65f7fbc0abf5b21e8f (10000 ETH)
Private Key: 0xdbda1821b80551c9d65939329250298aa3472ba22f6ea921c0cf5d620ea67b97

Account #9: 0xa0ee7a142d267c1f36714e4a8f75612f20a79720 (10000 ETH)
Private Key: 0x2a871d0798f97d79848a013d4936a73bf4cc922c825d33c1cf7073dff6d409c6

Account #10: 0xbcd4042de499d14e55001ccbb24a551f3b954096 (10000 ETH)
Private Key: 0xf214f2b2cd398c806f84e317254e0f0b801d0643303237d97a22a48e01628897

Account #11: 0x71be63f3384f5fb98995898a86b02fb2426c5788 (10000 ETH)
Private Key: 0x701b615bbdfb9de65240bc28bd21bbc0d996645a3dd57e7b12bc2bdf6f192c82

Account #12: 0xfabb0ac9d68b0b445fb7357272ff202c5651694a (10000 ETH)
Private Key: 0xa267530f49f8280200edf313ee7af6b827f2a8bce2897751d06a843f644967b1

Account #13: 0x1cbd3b2770909d4e10f157cab84c7264073c9ec (10000 ETH)
Private Key: 0x47c99abed3324a2707c28affff1267e45918ec8c3f20b8aa892e8b065d2942dd

```

Localhost 8545

Account 3

0xf39...2266

9999.9994 ETH

Buy Send Swap

Assets Activity

Send

Nov 23 · To: 0x597...b8f7

-10 ETH

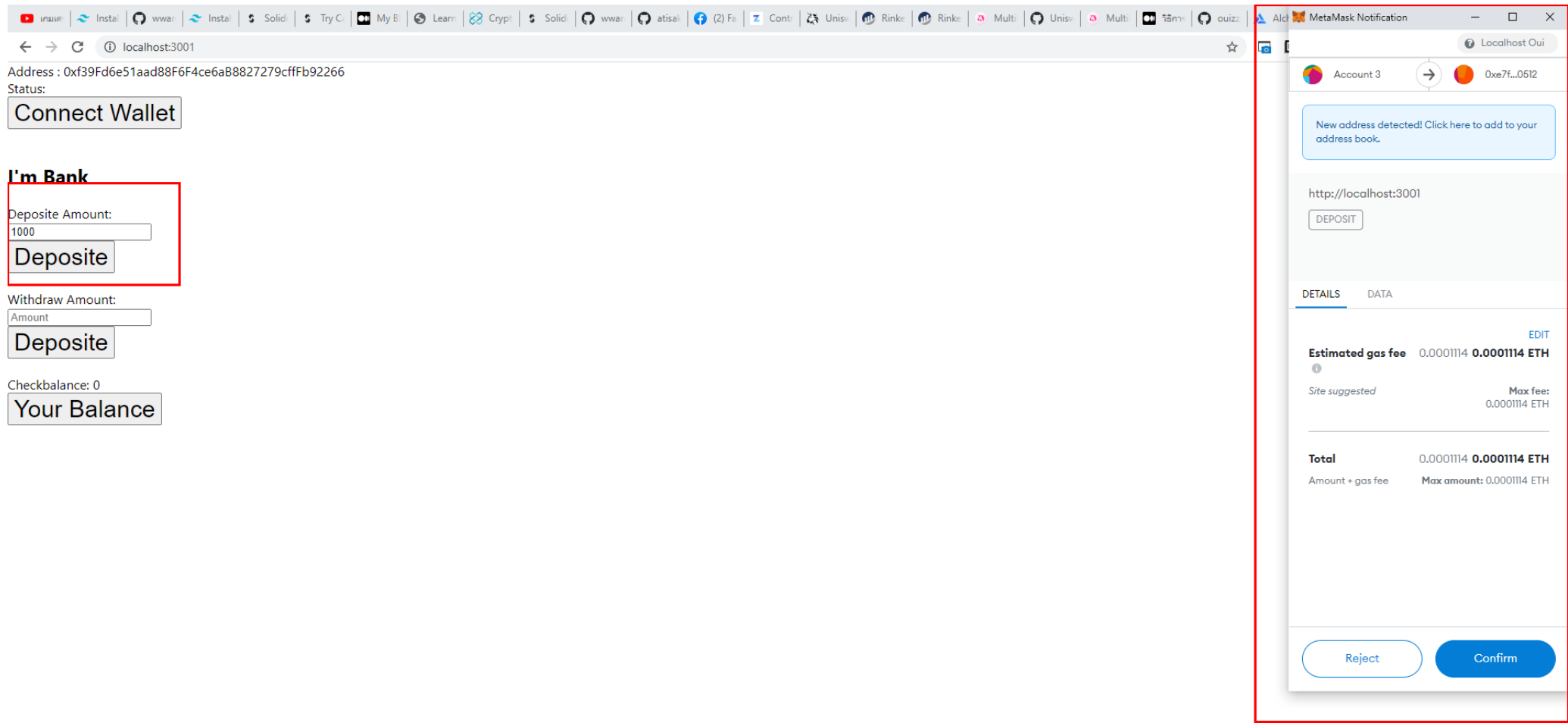
Give Right To Vote

Nov 23 · localhost:3000

-0 ETH

Connect Metamask by using localhost test

Bank Project



Deposit, enter the value into the input, it will show the details of the gas value and press confirm.

Bank Project

Address : 0xf39Fd6e51aad88F6F4ce6aB8827279cFfB92266

Status: Deposit already

Connect Wallet

I'm Bank

Deposit Amount:

1000

Deposit

Withdraw Amount:

Amount

Deposit

Checkbalance: 1000

Your Balance

Check the balance will show the deposited amount.
This section does not waste gas because it is a function view.

Blockchain Project

Chainlink Price Feed

Chainlink Price Feed

The image shows a Visual Studio Code editor with a TypeScript file named `ChainlinkPriceFeed.js` and a terminal window displaying the test results.

ChainlinkPriceFeed.js

```
1 const { expect } = require("chai");
2 const { ethers } = require("hardhat");
3
4 describe("Chainlink Price Feed", function () {
5
6   it("Get LINK price", async () => {
7     const PriceFeed = await ethers.getContractFactory("ChainlinkPriceFeed");
8     const priceFeed = await PriceFeed.deploy();
9
10    await priceFeed.deployed();
11
12    const output = await priceFeed.getCurrentPrice();
13    const outputETH = await priceFeed.getCurrentPriceETH();
14
15    console.log("OUTPUT BTC :", (Number(output) / 100000000) // 8 decimal places
16    console.log("OUTPUT ETH :", (Number(outputETH) / 100000000) // 8 decimal places
17
18    expect(Number(output)).to.not.equal(0)
19  });
20
21 });
22
```

Terminal Output

```
MINGW64/d/chainlink_workshop/simple-dex
$ npx hardhat test test/ChainlinkPriceFeed.js

Chainlink Price Feed
OUTPUT BTC : 63354.53
OUTPUT ETH : 4138.81
1) Get LINK price

0 passing (20s)
1 failing

1) Chainlink Price Feed
  Get LINK price:
    Error: Timeout of 20000ms exceeded. For async tests and hooks, ensure "done()" is called; if returning a Promise, ensure it resolves. (D:\chainlink_workshop\simple-dex\test\ChainlinkPriceFeed.js)
    at processTicksAndRejections (node:internal/process/task_queues:96:5)

MINGW64/d/chainlink_workshop/simple-dex
$ npx hardhat test test/ChainlinkPriceFeed.js

Chainlink Price Feed
OUTPUT BTC : 63354.53
OUTPUT ETH : 4138.81
✓ Get LINK price (12440ms)

1 passing (12s)

MINGW64/d/chainlink_workshop/simple-dex
$
```

This section will use the chainlink to feed the currency value of each coin as shown BTC and ETH currency to show.