

Relazione del progetto di Programmazione di reti

Juri Simoncini 0000922072

Luglio 2022

Indice

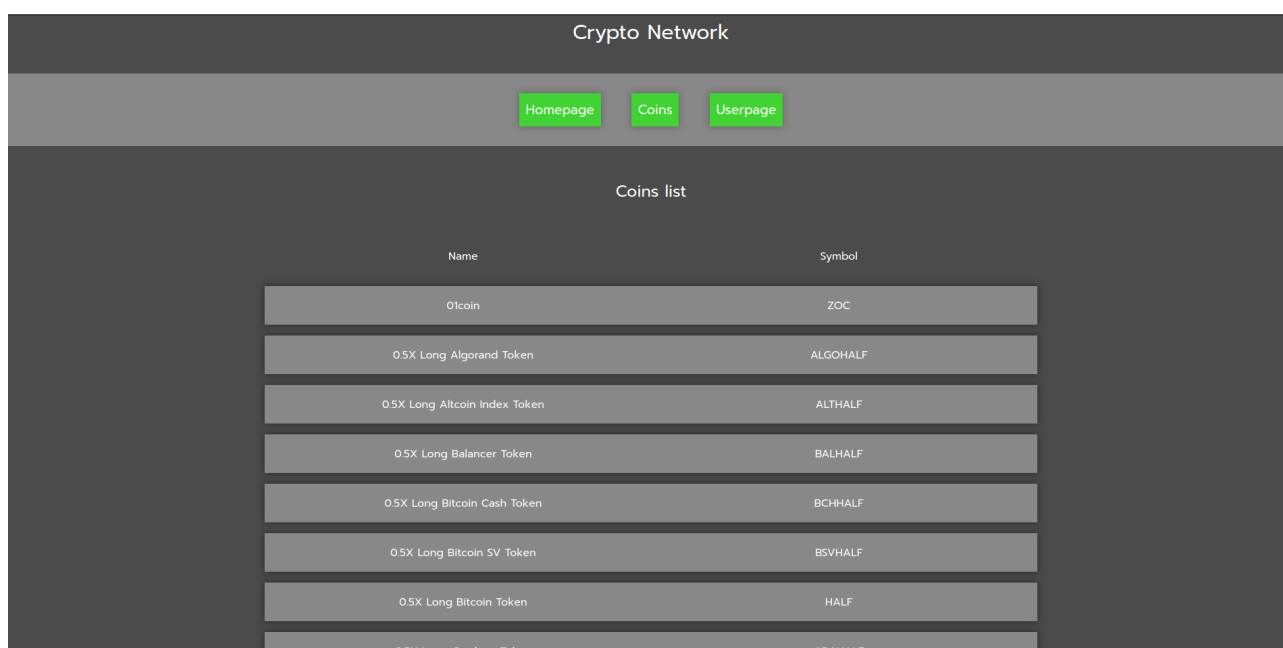
1	Introduzione	3
2	Pagina Principale	4
3	Singola valuta	5
4	Utente	6
5	Architettura del server	7

1 Introduzione

Il progetto ricrea una piattaforma di compravendita di criptovalute.
Il server per avere dati reali sulle cripto sfrutta l' API di www.coingecko.com che ho già utilizzato per altri progetti.

2 Pagina principale

All'avvio il server stampa su console l'indirizzo su cui è in ascolto. Una volta effettuata la richiesta da browser il server risponde visualizzando la homepage che poi è la pagina di lista di criptovalute. Cliccando su una crypto scatta una richiesta GET. La query string permette di generare la giusta pagina di informazioni della crypto selezionata.



The screenshot shows a web application titled "Crypto Network". It has a navigation bar with three buttons: "Homepage", "Coins", and "Userpage". Below the navigation bar, the page is titled "Coins list" and displays a table of cryptocurrencies. The table has two columns: "Name" and "Symbol". The table lists several cryptocurrencies, including Otcoin, 05X Long Algorand Token, 05X Long Altcoin Index Token, 05X Long Balancer Token, 05X Long Bitcoin Cash Token, 05X Long Bitcoin SV Token, 05X Long Bitcoin Token, and 05X Long Cardano Token.

Name	Symbol
Otcoin	ZOC
05X Long Algorand Token	ALGOHALF
05X Long Altcoin Index Token	ALTHALF
05X Long Balancer Token	BALHALF
05X Long Bitcoin Cash Token	BCHHALF
05X Long Bitcoin SV Token	BSVHALF
05X Long Bitcoin Token	HALF
05X Long Cardano Token	ADAHALF

Figura 1 : home page

3 Singola valuta

Appena arriva la richiesta al server questo genera la pagina con i dettagli della cripto recuperati da coingecko.

Una volta dentro la pagina appare nella navbar una voce buy/sell che ci permette di acquistare vendere cripto se l'utente è autorizzato dal server

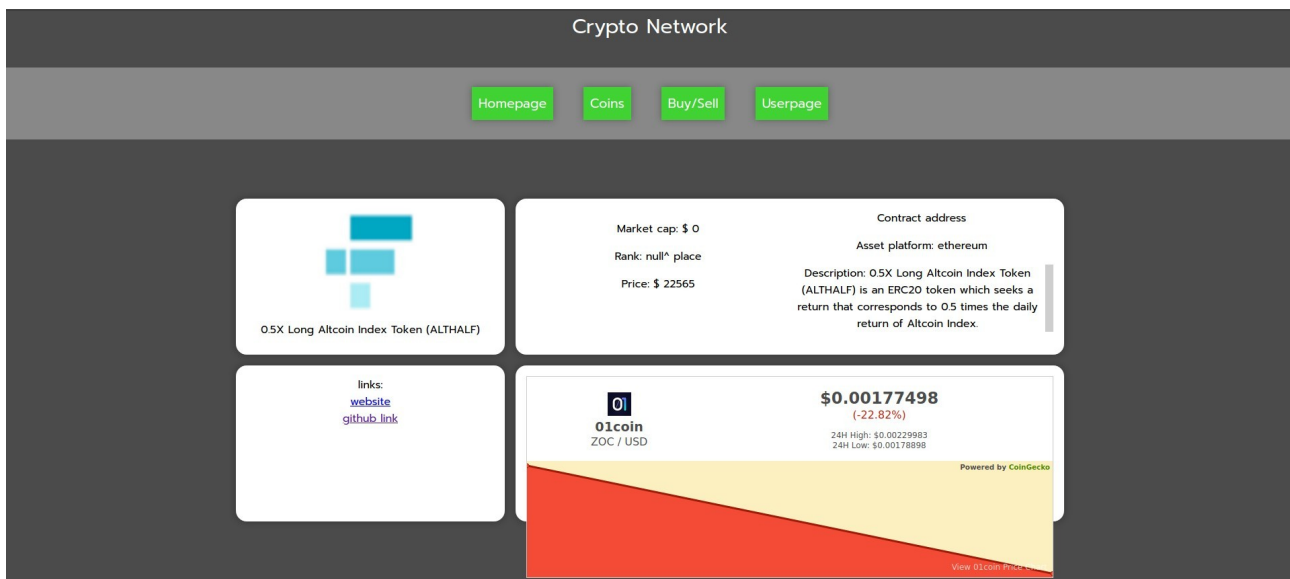


Figura 2: cripto info

4 Utente

La piattaforma per questioni di semplicità non permette la registrazione di nuovi utenti ma vengono forniti account di prova nel file “users.json”. La pagina utente si presenta con: nome utente, bilancio in USD, wallet dell'utente.

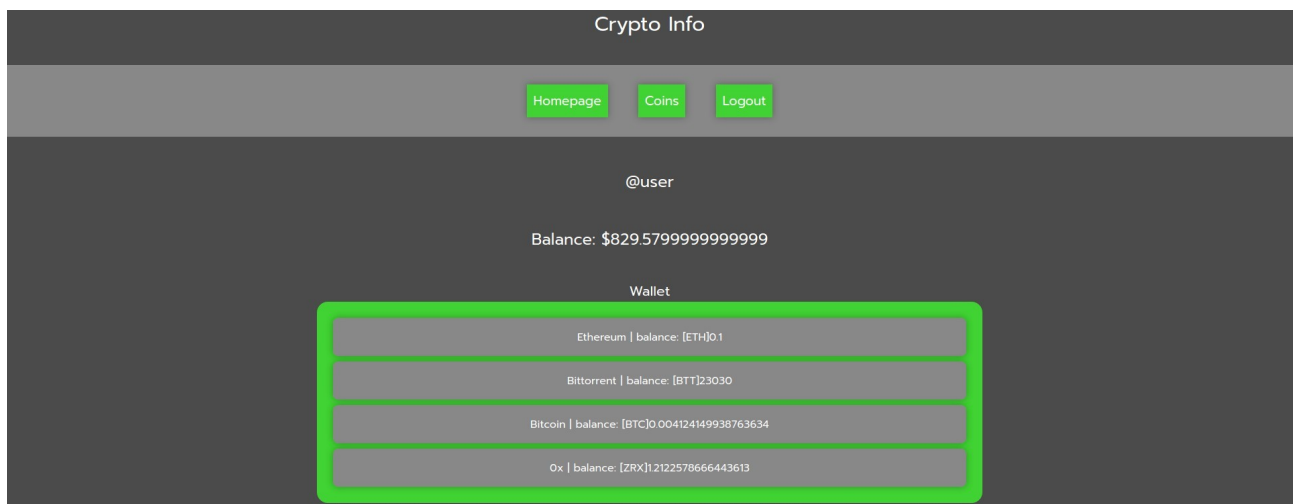


Figura 3: Pagina utente

5 Architettura del server

Il server riceve, risponde e gestisce le richieste in arrivo dal client (web browser).

Librerie usate:

1. **JSON**: utilizzando molti oggetti json la libreria json mi permette di effettuare il parsing di oggetti semplici a json
2. **CGI**: utilizzata per parsing di oggetti una request body di una richiesta POST
3. **URLIB**: utilizzata per ottenere data in query string di richieste GET

La classe ReqHandle implementa i metodi di funzionamento del server, sono presenti due metodi:

`guess_get_request()`: cerca il percorso presente nell'header http della richiesta e ritorna una risposta di tipo json o html.

`guess_post_request()`: cerca il percorso presente nell'header http della richiesta e ritorna una risposta di tipo json.

Alcune pagine presenti nell'applicazione vengono generate nel momento della richiesta con i valori dati dalla richiesta stessa.
La generazione della pagina è effettuata grazie ad alcune utility presenti nella cartella util.

