KRYPC ASSIGNEMENT

Surya Suresh

INDEX

Contents

List Of Figure

1. Introduction

1.1 Aim: Create a script that retrieves the current Ether price from an API such as CoinGecko or CryptoCompare and displays it on a web page. The script should also display the price in different fiat currencies.

1.2 Terminology

API = mechanisms that enable two software components to communicate with each other using a set of definitions and protocol

Standard HTTP methods = GET(getting data), POST(creating data), PUT (updating data),DELETE (Deleting data)

1.3 Comparison

This table below shows the comparison between CoinGecko and CryptoCompare features

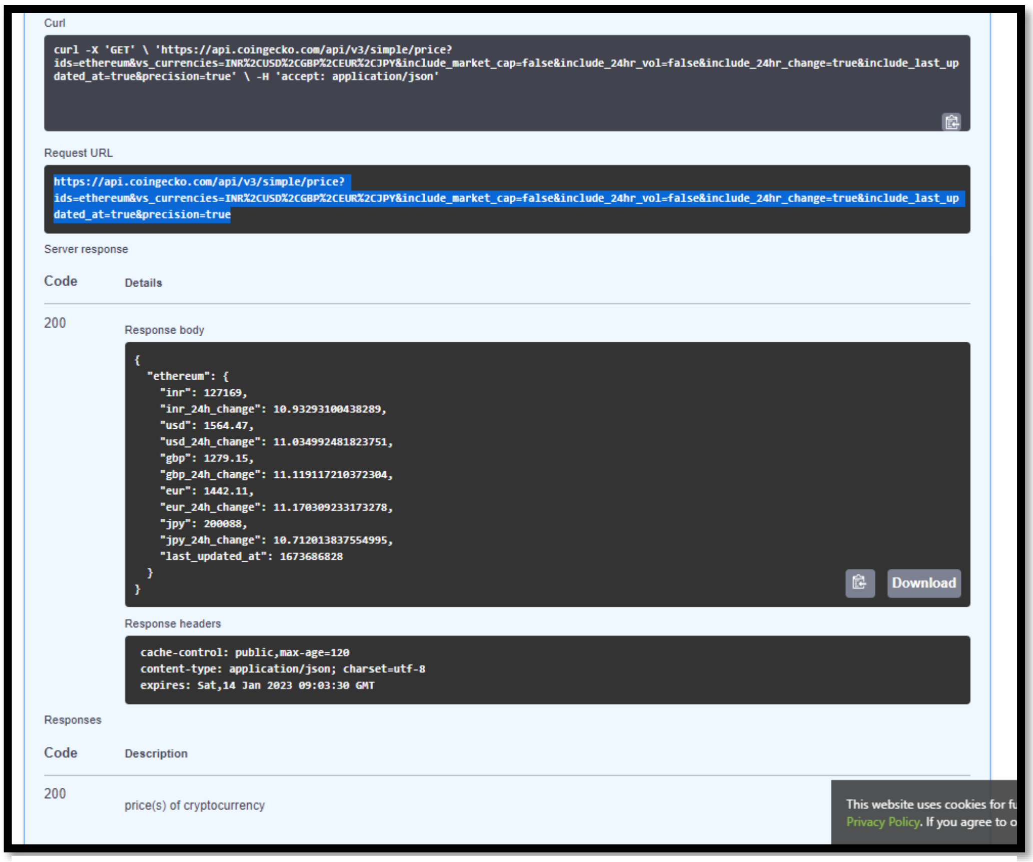
	CoinGecko	CryptoCompare
1	Free API of CoinGecko has a rate limit of 10-50 calls per minute, but doesn't have a total API limit. If you exceed that limit you will be blocked for the next 1 minute window.	CryptoCompare's API is limited to 2000 data points per call for personal & non-commercial projects. Capped at 250,000 lifetime API calls.
2	CoinGecko has free and paid API service. But when it comes to free version it has more features compared to CryptoCompare.	CryptoCompare's has free and paid API service. Free API service has limitations.
3	CoinGecko free version Endpoints are all cached to around 1 to 5 minutes and you can expect most data to be updated at similar intervals. But Pro API(paid plans) generally have faster update frequency, i.e. 30 sec for simple/price endpoint.	CryptoCompare data updation is realtime which make it faster compared to Coingecko for retrieving data.

Other Websites that provides free API services for cryptocurrency:

1. Alpha Vantage = https://www.alphavantage.co/documentation/
2. Fixer Currency = https://rapidapi.com/fixer/api/fixer-currency/
3. Mineable Coins = https://api.minerstat.com/docs-coins/documentation

2. Procedures performed

2.1 APIs for Ethereum price:


Curl

```
curl -X 'GET' \ 'https://api.coingecko.com/api/v3/simple/price?
ids=ethereum&vs_currencies=INR%2CUSD%2CGBP%2CEUR%2CJPY&include_market_cap=false&include_24hr_vol=false&include_24hr_change=true&include_last_up
dated_at=true&precision=true' \ -H 'accept: application/json'
```

Request URL

```
https://api.coingecko.com/api/v3/simple/price?
ids=ethereum&vs_currencies=INR%2CUSD%2CGBP%2CEUR%2CJPY&include_market_cap=false&include_24hr_vol=false&include_24hr_change=true&include_last_up
dated_at=true&precision=true
```

Server response

Code	Details
200	Response body

```
{
  "ethereum": {
    "inr": 127169,
    "inr_24h_change": 10.93293100438289,
    "usd": 1564.47,
    "usd_24h_change": 11.034992481823751,
    "gbp": 1279.15,
    "gbp_24h_change": 11.119117210372304,
    "eur": 1442.11,
    "eur_24h_change": 11.170309233173278,
    "jpy": 200088,
    "jpy_24h_change": 10.712013837554995,
    "last_updated_at": 1673686828
  }
}
```

Response headers

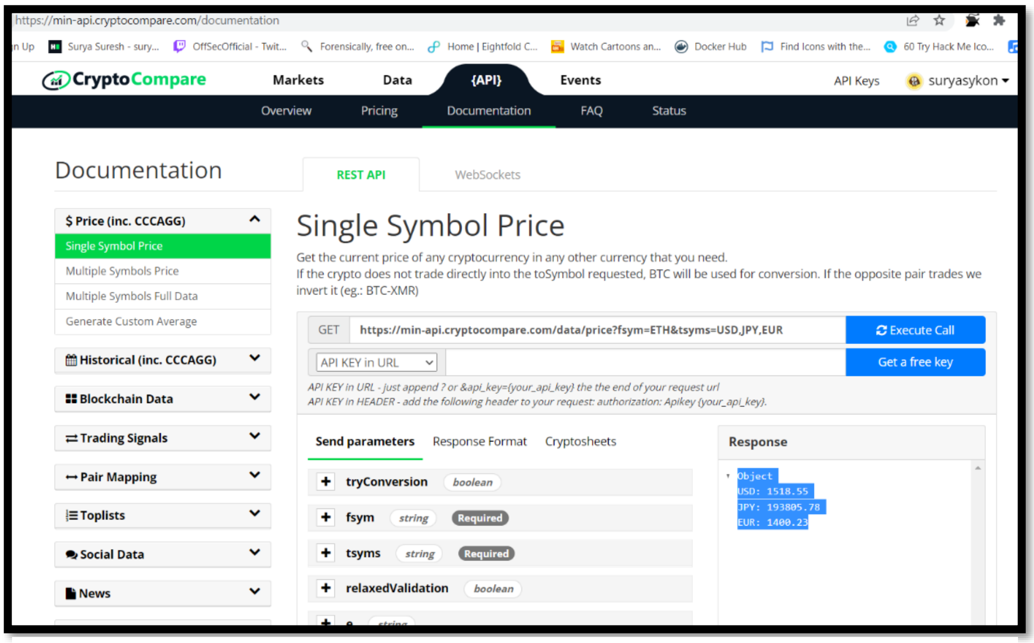
```
cache-control: public,max-age=120
content-type: application/json; charset=utf-8
expires: Sat,14 Jan 2023 09:03:30 GMT
```

Responses

Code	Description
200	price(s) of cryptocurrency

This website uses cookies for fu
Privacy Policy. If you agree to o

Figure 1: CoinGecko API execution


https://min-api.cryptocompare.com/documentation

Documentation

Single Symbol Price

Get the current price of any cryptocurrency in any other currency that you need.
If the crypto does not trade directly into the toSymbol requested, BTC will be used for conversion. If the opposite pair trades we invert it (eg.: BTC-XMR)

GET https://min-api.cryptocompare.com/data/price?fsym=ETH&tsyms=USD,JPY,EUR

API KEY in URL

API KEY in URL - just append ? or &api_key={your_api_key} the the end of your request url
API KEY in HEADER - add the following header to your request: authorization: Apikey {your_api_key}.

tryConversion	boolean	
fsym	string	Required
tsyms	string	Required
relaxedValidation	boolean	

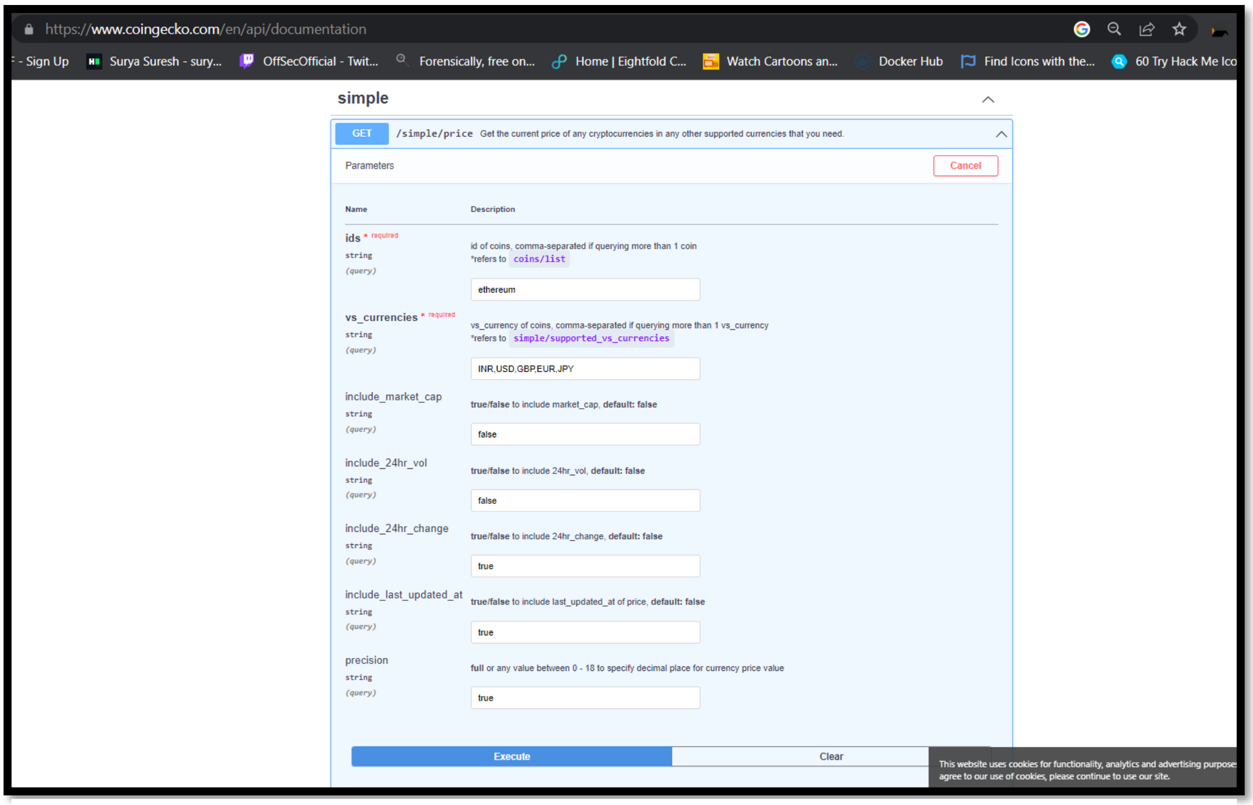
Response

```
Object
USD: 1518.55
JPY: 193805.78
EUR: 1400.23
```

Figure 2: CryptoCompare API execution

2.2 Getting API from coingecko.

I. Coingecko has various GET methods such as including market cap, 24hr_vol, historical data (name, price, market, stats). But my task was to retrieves the current Ehereum price from an API such as CoinGecko



simple

GET /simple/price Get the current price of any cryptocurrencies in any other supported currencies that you need.

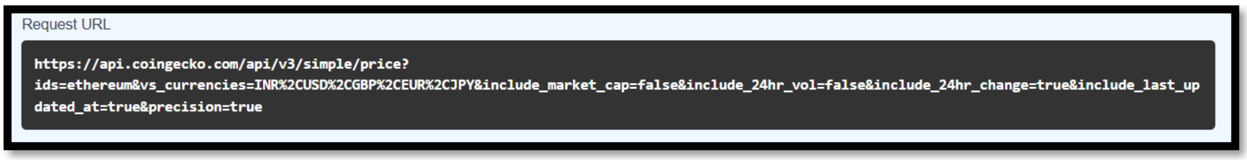
Parameters

Name	Description
ids * required string (query)	id of coins, comma-separated if querying more than 1 coin *refers to `coins/list` — ethereum
vs_currencies * required string (query)	vs_currency of coins, comma-separated if querying more than 1 vs_currency *refers to `simple/supported_vs_currencies` — INR,USD,GBP,EUR,JPY
include_market_cap string (query)	true/false to include market_cap, **default: false** — false
include_24hr_vol string (query)	true/false to include 24hr_vol, **default: false** — false
include_24hr_change string (query)	true/false to include 24hr_change, **default: false** — true
include_last_updated_at string (query)	true/false to include last_updated_at of price, **default: false** — true
precision string (query)	full or any value between 0 - 18 to specify decimal place for currency price value — true

This website uses cookies for functionality, analytics and advertising purpose agree to our use of cookies, please continue to use our site.

Figure 3: CoinGecko Userinterface

II. After executing it generates Request URL

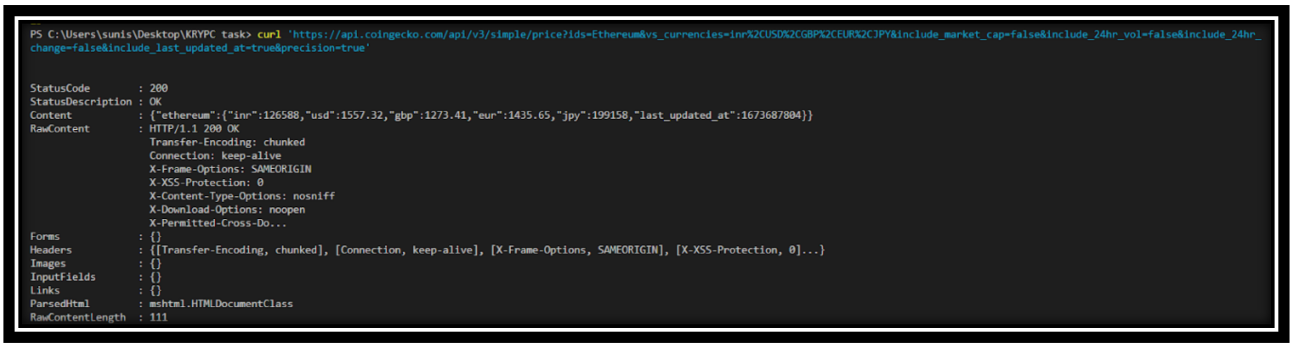


Request URL

```
https://api.coingecko.com/api/v3/simple/price?
ids=ethereum&vs_currencies=INR%2CUSD%2CGBP%2CEUR%2CJPY&include_market_cap=false&include_24hr_vol=false&include_24hr_change=true&include_last_up
dated_at=true&precision=true
```

Figure 4: Coingecko API

III. Output. So the data is in json format.

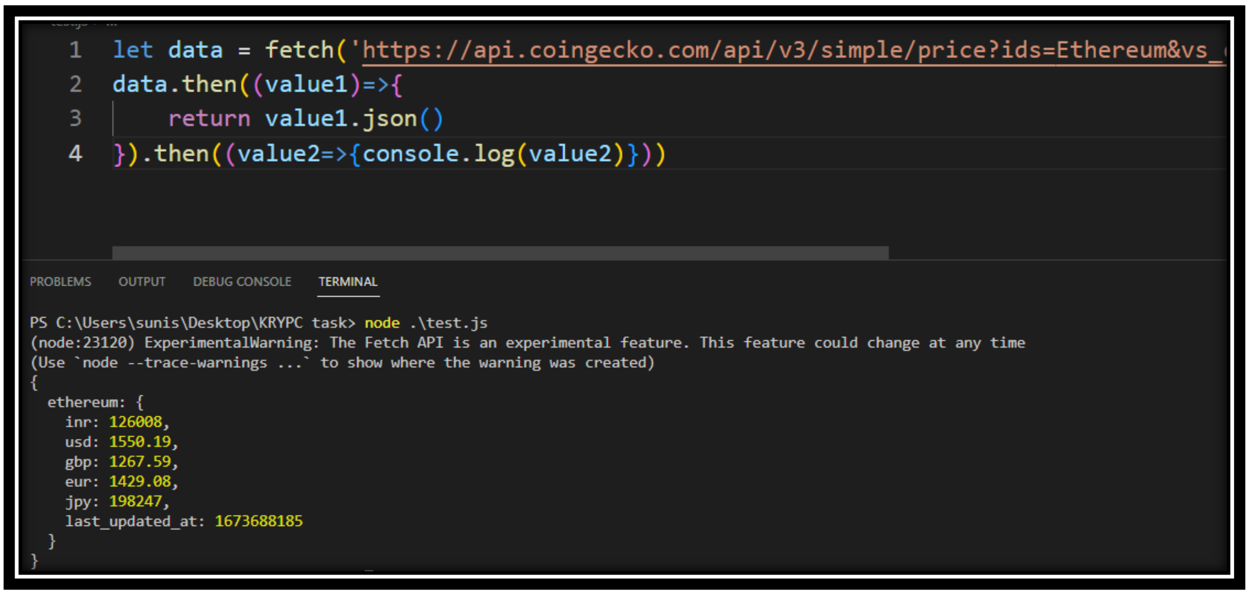


```
PS C:\Users\sunis\Desktop\KRYPC task> curl 'https://api.coingecko.com/api/v3/simple/price?ids=Ethereum&vs_currencies=inr%2CUSD%2CGBP%2CEUR%2CJPY&include_market_cap=false&include_24hr_vol=false&include_24hr_
change=false&include_last_updated_at=true&precision=true'


StatusCode        : 200
StatusDescription : OK
Content           : {"ethereum":{"inr":126588,"usd":1557.32,"gbp":1273.41,"eur":1435.65,"jpy":199158,"last_updated_at":1673687804}}
RawContent        : HTTP/1.1 200 OK
                    Transfer-Encoding: chunked
                    Connection: keep-alive
                    X-Frame-Options: SAMEORIGIN
                    X-XSS-Protection: 0
                    X-Content-Type-Options: nosniff
                    X-Download-Options: noopen
                    X-Permitted-Cross-Do...
Forms             : {}
Headers           : {[Transfer-Encoding, chunked], [Connection, keep-alive], [X-Frame-Options, SAMEORIGIN], [X-XSS-Protection, 0]...}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : mshtml.HTMLDocumentClass
RawContentLength  : 111
```

Figure 5: retrieving data in json format

2.3 Creating logic for retrieving current Ethereum price

I. Here using the fetch method retrieves the values.



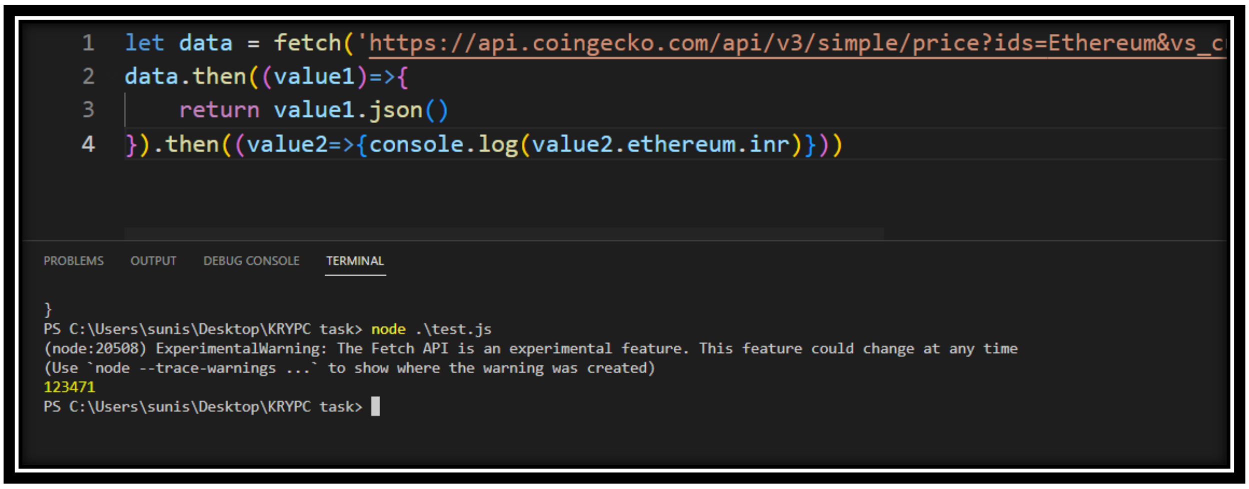
```
let data = fetch('https://api.coingecko.com/api/v3/simple/price?ids=Ethereum&vs_
data.then((value1)=>{
    return value1.json()
}).then((value2=>{console.log(value2)}))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\sunis\Desktop\KRYPC task> node .\test.js
(node:23120) ExperimentalWarning: The Fetch API is an experimental feature. This feature could change at any time
(Use `node --trace-warnings ...` to show where the warning was created)
{
  ethereum: {
    inr: 126008,
    usd: 1550.19,
    gbp: 1267.59,
    eur: 1429.08,
    jpy: 198247,
    last_updated_at: 1673688185
  }
}
```

Figure 6: Fetch Method

II. Now to access a particular in the object i used ‘ . ’ notation .
E.g. here Ethereum price in INR



```
let data = fetch('https://api.coingecko.com/api/v3/simple/price?ids=Ethereum&vs_c
data.then((value1)=>{
    return value1.json()
}).then((value2=>{console.log(value2.ethereum.inr)}))
```

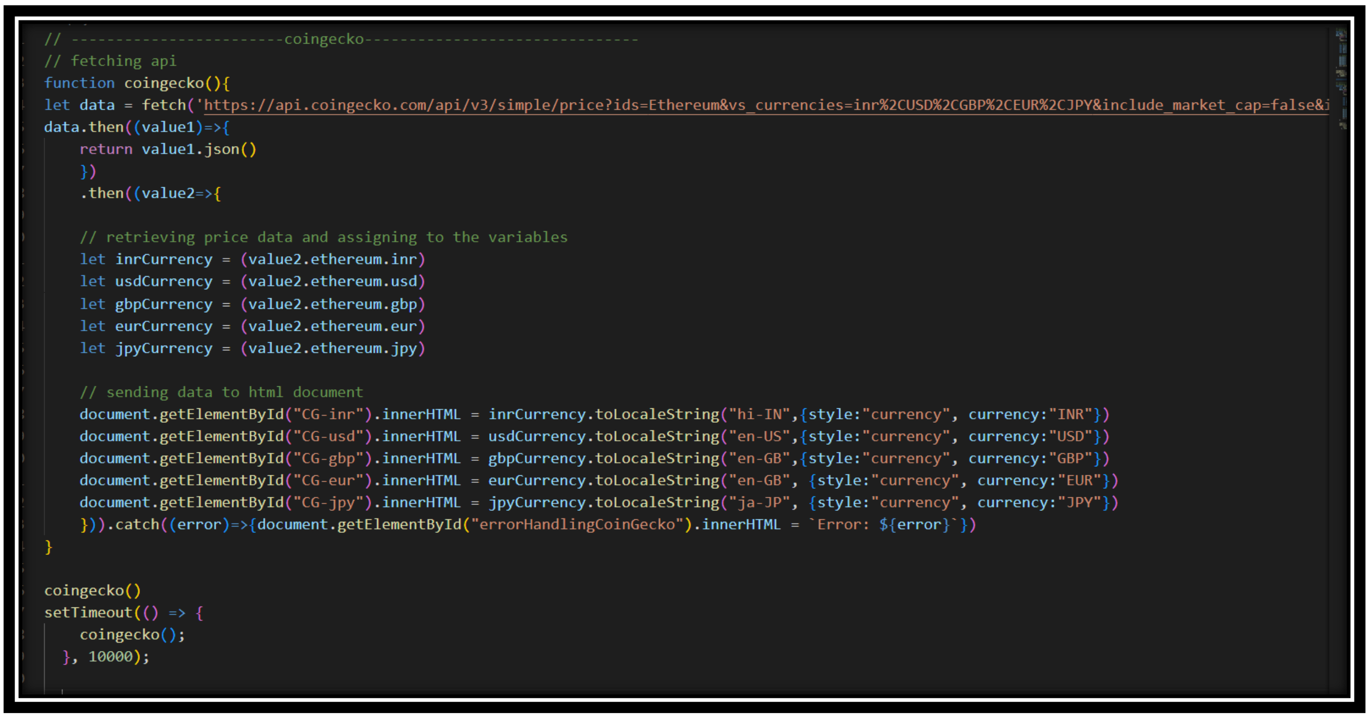
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
}
PS C:\Users\sunis\Desktop\KRYPC task> node .\test.js
(node:20508) ExperimentalWarning: The Fetch API is an experimental feature. This feature could change at any time
(Use `node --trace-warnings ...` to show where the warning was created)
123471
PS C:\Users\sunis\Desktop\KRYPC task> 
```

Figure 7: Accessing objects

III. Here's the complete logic

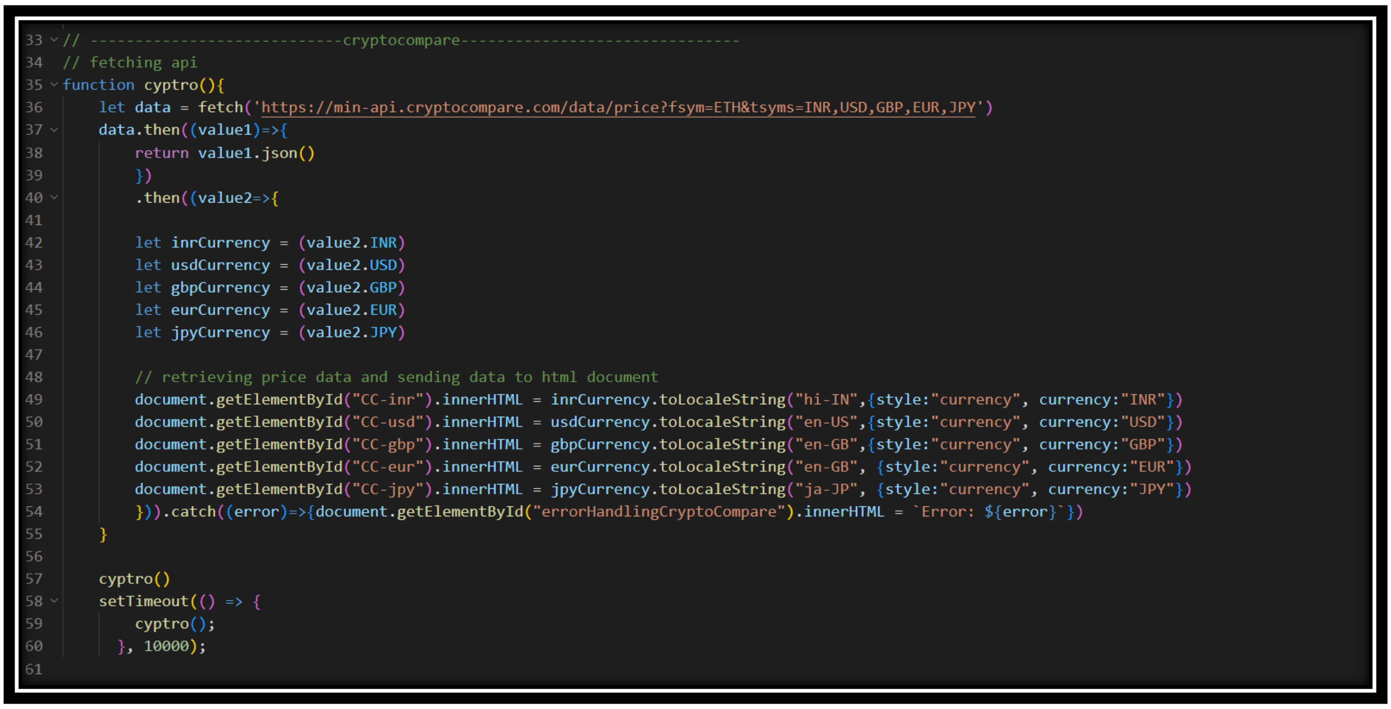
A. logic For CoinGecko



```
// -----------------------coingecko------------------------------
// fetching api
function coingecko(){
let data = fetch('https://api.coingecko.com/api/v3/simple/price?ids=Ethereum&vs_currencies=inr%2CUSD%2CGBP%2CEUR%2CJPY&include_market_cap=false&i
data.then((value1)=>{
    return value1.json()
    })
    .then((value2=>{

    // retrieving price data and assigning to the variables
    let inrCurrency = (value2.ethereum.inr)
    let usdCurrency = (value2.ethereum.usd)
    let gbpCurrency = (value2.ethereum.gbp)
    let eurCurrency = (value2.ethereum.eur)
    let jpyCurrency = (value2.ethereum.jpy)

    // sending data to html document
    document.getElementById("CG-inr").innerHTML = inrCurrency.toLocaleString("hi-IN",{style:"currency", currency:"INR"})
    document.getElementById("CG-usd").innerHTML = usdCurrency.toLocaleString("en-US",{style:"currency", currency:"USD"})
    document.getElementById("CG-gbp").innerHTML = gbpCurrency.toLocaleString("en-GB",{style:"currency", currency:"GBP"})
    document.getElementById("CG-eur").innerHTML = eurCurrency.toLocaleString("en-GB", {style:"currency", currency:"EUR"})
    document.getElementById("CG-jpy").innerHTML = jpyCurrency.toLocaleString("ja-JP", {style:"currency", currency:"JPY"})
    })).catch((error)=>{document.getElementById("errorHandlingCoinGecko").innerHTML = `Error: ${error}`})
}

coingecko()
setTimeout(() => {
    coingecko();
  }, 10000);
```

Figure 8: logic For CoinGecko API

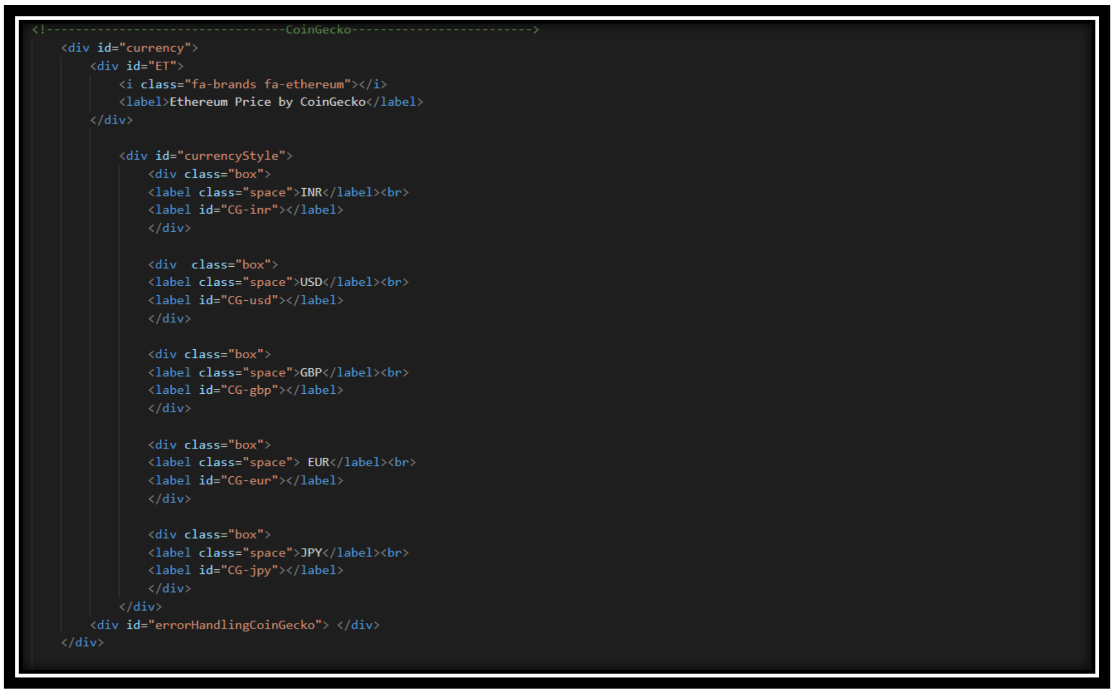
B. logic For CryptoCompare



```
// --------------------------cryptocompare-----------------------------
// fetching api
function cyptro(){
    let data = fetch('https://min-api.cryptocompare.com/data/price?fsym=ETH&tsyms=INR,USD,GBP,EUR,JPY')
    data.then((value1)=>{
        return value1.json()
        })
        .then((value2=>{

        let inrCurrency = (value2.INR)
        let usdCurrency = (value2.USD)
        let gbpCurrency = (value2.GBP)
        let eurCurrency = (value2.EUR)
        let jpyCurrency = (value2.JPY)

        // retrieving price data and sending data to html document
        document.getElementById("CC-inr").innerHTML = inrCurrency.toLocaleString("hi-IN",{style:"currency", currency:"INR"})
        document.getElementById("CC-usd").innerHTML = usdCurrency.toLocaleString("en-US",{style:"currency", currency:"USD"})
        document.getElementById("CC-gbp").innerHTML = gbpCurrency.toLocaleString("en-GB",{style:"currency", currency:"GBP"})
        document.getElementById("CC-eur").innerHTML = eurCurrency.toLocaleString("en-GB", {style:"currency", currency:"EUR"})
        document.getElementById("CC-jpy").innerHTML = jpyCurrency.toLocaleString("ja-JP", {style:"currency", currency:"JPY"})
        })).catch((error)=>{document.getElementById("errorHandlingCryptoCompare").innerHTML = `Error: ${error}`})
    }

    cyptro()
    setTimeout(() => {
        cyptro();
      }, 10000);

```

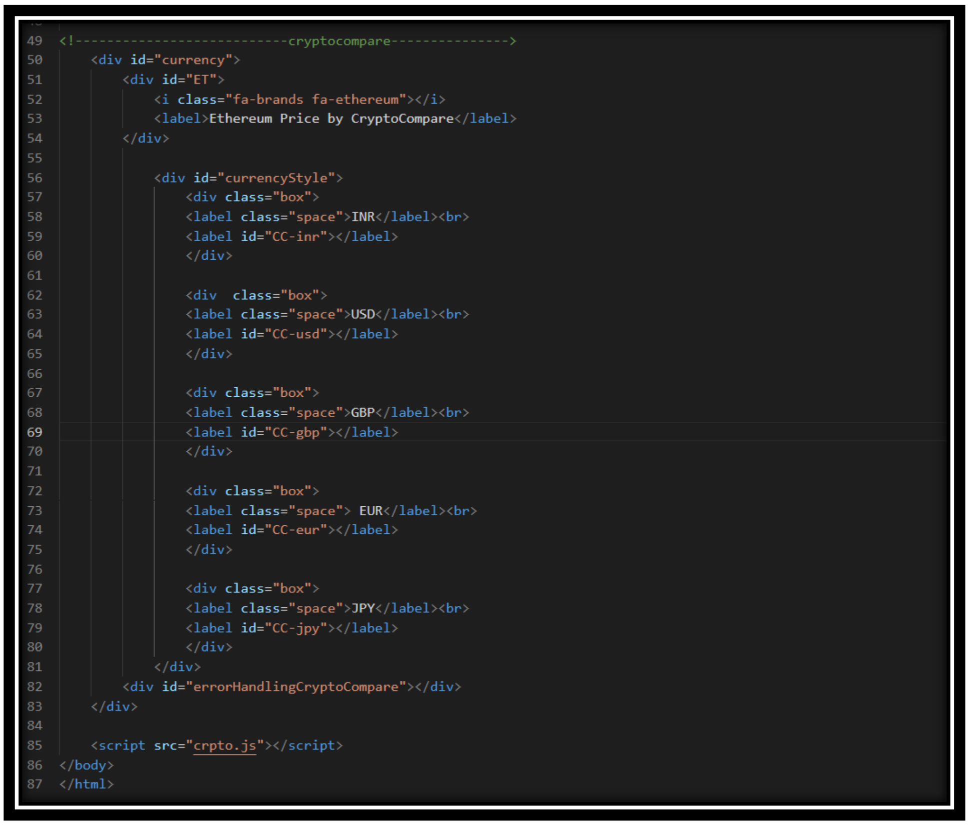
Figure 9: logic For CryptoCompare

2.4 Creating HTML and adding CSS



```html
<!--------------------------------CoinGecko----------------------->
    <div id="currency">
        <div id="ET">
            <i class="fa-brands fa-ethereum"></i>
            <label>Ethereum Price by CoinGecko</label>
        </div>

            <div id="currencyStyle">
                <div class="box">
                <label class="space">INR</label><br>
                <label id="CG-inr"></label>
                </div>

                <div  class="box">
                <label class="space">USD</label><br>
                <label id="CG-usd"></label>
                </div>

                <div class="box">
                <label class="space">GBP</label><br>
                <label id="CG-gbp"></label>
                </div>

                <div class="box">
                <label class="space"> EUR</label><br>
                <label id="CG-eur"></label>
                </div>

                <div class="box">
                <label class="space">JPY</label><br>
                <label id="CG-jpy"></label>
                </div>
            </div>
        <div id="errorHandlingCoinGecko"> </div>
    </div>
```

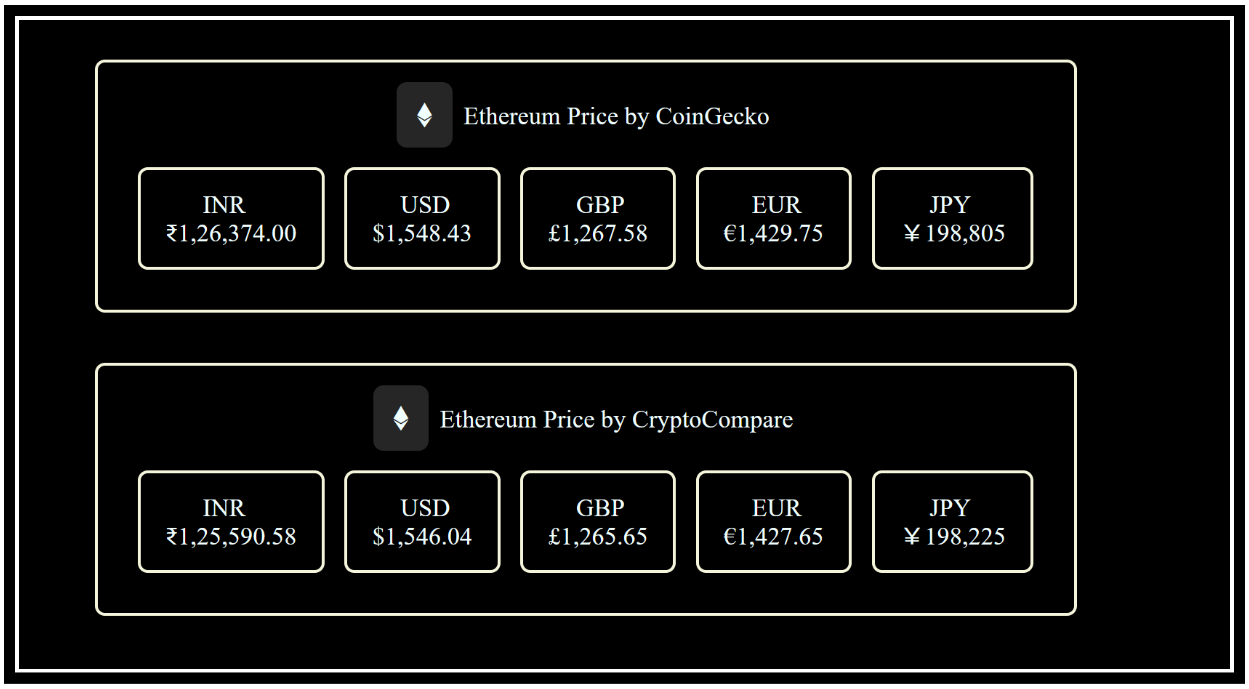
Figure 10: HTMLfor CoinGecko



```html
<!--------------------------cryptocompare-------------->
    <div id="currency">
        <div id="ET">
            <i class="fa-brands fa-ethereum"></i>
            <label>Ethereum Price by CryptoCompare</label>
        </div>

            <div id="currencyStyle">
                <div class="box">
                <label class="space">INR</label><br>
                <label id="CC-inr"></label>
                </div>

                <div  class="box">
                <label class="space">USD</label><br>
                <label id="CC-usd"></label>
                </div>

                <div class="box">
                <label class="space">GBP</label><br>
                <label id="CC-gbp"></label>
                </div>

                <div class="box">
                <label class="space"> EUR</label><br>
                <label id="CC-eur"></label>
                </div>

                <div class="box">
                <label class="space">JPY</label><br>
                <label id="CC-jpy"></label>
                </div>
            </div>
        <div id="errorHandlingCryptoCompare"></div>
    </div>

    <script src="crpto.js"></script>
</body>
</html>
```

Figure 11 HTML for CryptoCompare

2.5 OUTPUT :



Ethereum Price by CoinGecko

INR ₹1,26,374.00
USD $1,548.43
GBP £1,267.58
EUR €1,429.75
JPY ¥ 198,805

Ethereum Price by CryptoCompare

INR ₹1,25,590.58
USD $1,546.04
GBP £1,265.65
EUR €1,427.65
JPY ¥ 198,225

Figure 12: Output of CoinGecko and CryptoCompare