# PROPOSAL FOR AUTO SPOT TRADING BOT

***NOTE - This proposal will be submitted for voting in 7 days, which is the 27th of March, 2023. Please leave your comments, suggestions and objections on the official Agora forum post before this date.***

## CURRENT FEATURES

-- Completely original code written from scratch (not copied from another open-source project)

-- Supports CEX, DEX and other swap pools for trades

-- Makes 50-300 trades in a day

-- Protects assets from sudden price fluctuations by monitoring currencies & selling early

-- No code & No configuration required

-- Configured for short term trading 24x7 without user intervention

-- Does not use the native swap module (does not require access to LUNC's swap pool)

-- Supports swaps (not orders) for trades to make instantaneous trades without significant slippage

-- Supports spot trading which is much safer than margin trading

-- ROI (according to calculations) is 750-5,000% per year

## FUTURE INTEGRATIONS

-- Restore/fix LUNC swap pool and core swap module

-- Configure bot to use LUNC ecosystem tokens/coins for swap only

-- Support USTC as the default gas token

## What is an auto spot trading bot?

Spot trading is simple trading in which orders are realized at current prices.

An auto trading bot for spot trading is a software solution to automate the process of trading, relying upon historical data which is processed and analyzed for meaningful KPIs, which help in making a financial decision on spot.

A spot trade can either be a buy or a sell. It can also be a swap between currencies.

## Doesn't Binance and Kucoin have auto spot trading bots?

Yes, but they utilize a process called Grid Trading in which the bot creates a grid of orders at the prices that the user specifies according to certain criteria.

The issue with grid trading is that if prices do not follow a regular pattern (and a smooth curve on the chart), it will not execute the lined-up orders which will remain pending. It would require advanced knowledge of trading and also user intervention to set up and use the bots. Most traders do not utilize such bots for these reasons.

The auto spot trading bot I have written only trades via swaps. It does not utilize orders which may or may not execute. We have completely removed such a scenario by utilizing swap pools, in which we can have instantaneous swaps between crypto coins and stable coins. Currently, it is using the Binance Swap Pool Trading API to get data for swap pairs and making swaps.

For example, the bot swaps USDT/BTC when the price of BTC is increasing, and then BTC/USDT when the price of BTC is decreasing. This ensures that trades are executed at exact prices, preventing loss due to sudden price fluctuations and pending orders.

## Does it support Margin Trading?

Currently, it only supports Spot Trading, which is much safer than Margin Trading (which utilizes collaterals and borrowed/loaned money to trade).

Margin Trading uses the principle of Buying Long and Selling Short, which is also integrated into the auto trading bot implicitly, but it is protected from over-collateralized assets since the bot uses Spot Trading only to make trades. It will automatically stop trading if the account balance is converted to fiat or falls below the required minimum balance to make a swap trade (usually $5-10 is the lower limit for a trade).

## Does it support Machine Learning and Artificial Intelligence?

No. The architecture of the bot has been designed to be instantaneous, reactive and responsive to the minute. ML & AI takes into consideration significant amounts of historical data which may prevent it from reacting to sudden price fluctuations in the market (as it is generally observed in the cryptocurrency industry). The bot works perfectly without the requirement to implement ML & AI immediately.

## How is this bot relevant to Terra Luna Classic?

Currently, the bot fetches cryptocurrency data from CoinGecko (primary) and CoinMarketCap (secondary), compares it to swap pool data from Binance Swap Pool Trading APIs, and makes an instantaneous trade using the same Binance Swap Pool Trading API. This allows us access to 90+ coins to make swap trades using (200+ swap pairs).

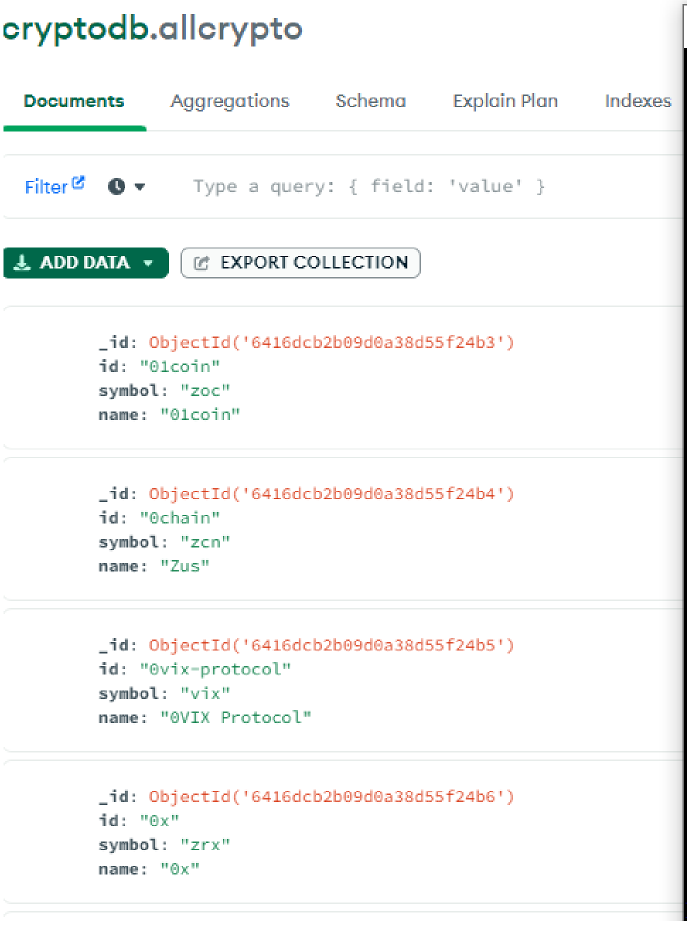
All the bot requires, is a swap pool to make trades. So, it can connect to CEX swap pools, DEX swap pools, as well as swap pools of individual cryptocurrencies, like Terra Luna Classic.

Since Terra Luna Classic already has a swap module, the auto trading bot can utilize the swap module to get swap pool pair(s) data and make the final swap trade using the same.

The bot makes 50-300 trades in a single day. Multiple users utilizing the bot will significantly improve current volume rate(s) which will help bring back much required confidence in all the coins part of the Terra Luna Classic ecosystem, apart from making profits comparable to the best positive PNL rates in the market.

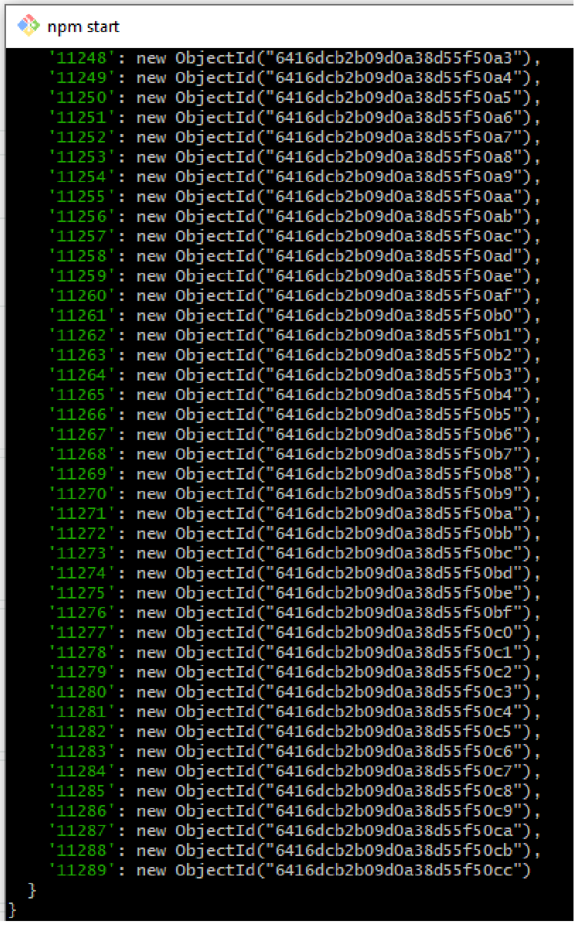
## How does the bot work?

1. Polls the CoinGecko/CoinMarketCap API(s) for the list of available cryptocurrencies:



cryptodb.allcrypto

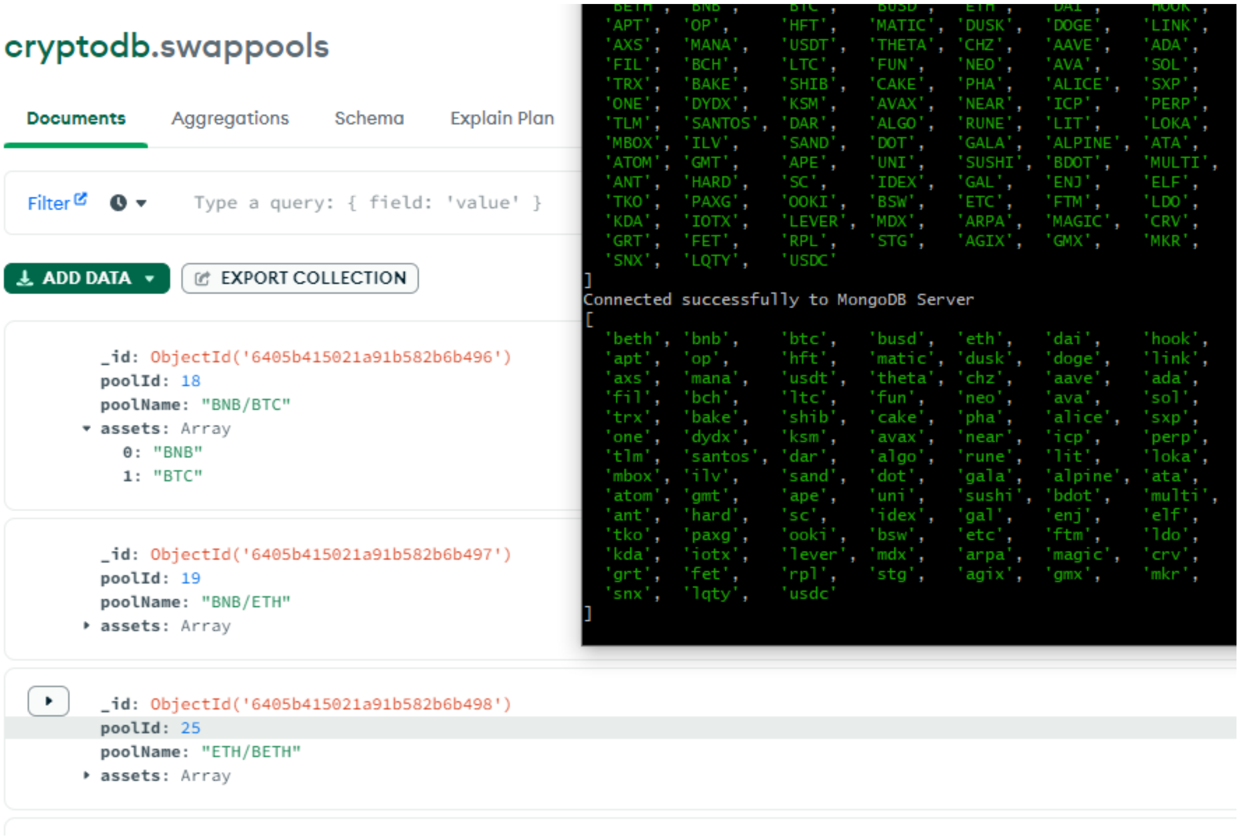
```
_id: ObjectId('6416dcb2b09d0a38d55f24b3')
id: "01coin"
symbol: "zoc"
name: "01coin"

_id: ObjectId('6416dcb2b09d0a38d55f24b4')
id: "0chain"
symbol: "zcn"
name: "Zus"

_id: ObjectId('6416dcb2b09d0a38d55f24b5')
id: "0vix-protocol"
symbol: "vix"
name: "0VIX Protocol"

_id: ObjectId('6416dcb2b09d0a38d55f24b6')
id: "0x"
symbol: "zrx"
name: "0x"
```



npm start

```
    '11248': new ObjectId("6416dcb2b09d0a38d55f50a3"),
    '11249': new ObjectId("6416dcb2b09d0a38d55f50a4"),
    '11250': new ObjectId("6416dcb2b09d0a38d55f50a5"),
    '11251': new ObjectId("6416dcb2b09d0a38d55f50a6"),
    '11252': new ObjectId("6416dcb2b09d0a38d55f50a7"),
    '11253': new ObjectId("6416dcb2b09d0a38d55f50a8"),
    '11254': new ObjectId("6416dcb2b09d0a38d55f50a9"),
    '11255': new ObjectId("6416dcb2b09d0a38d55f50aa"),
    '11256': new ObjectId("6416dcb2b09d0a38d55f50ab"),
    '11257': new ObjectId("6416dcb2b09d0a38d55f50ac"),
    '11258': new ObjectId("6416dcb2b09d0a38d55f50ad"),
    '11259': new ObjectId("6416dcb2b09d0a38d55f50ae"),
    '11260': new ObjectId("6416dcb2b09d0a38d55f50af"),
    '11261': new ObjectId("6416dcb2b09d0a38d55f50b0"),
    '11262': new ObjectId("6416dcb2b09d0a38d55f50b1"),
    '11263': new ObjectId("6416dcb2b09d0a38d55f50b2"),
    '11264': new ObjectId("6416dcb2b09d0a38d55f50b3"),
    '11265': new ObjectId("6416dcb2b09d0a38d55f50b4"),
    '11266': new ObjectId("6416dcb2b09d0a38d55f50b5"),
    '11267': new ObjectId("6416dcb2b09d0a38d55f50b6"),
    '11268': new ObjectId("6416dcb2b09d0a38d55f50b7"),
    '11269': new ObjectId("6416dcb2b09d0a38d55f50b8"),
    '11270': new ObjectId("6416dcb2b09d0a38d55f50b9"),
    '11271': new ObjectId("6416dcb2b09d0a38d55f50ba"),
    '11272': new ObjectId("6416dcb2b09d0a38d55f50bb"),
    '11273': new ObjectId("6416dcb2b09d0a38d55f50bc"),
    '11274': new ObjectId("6416dcb2b09d0a38d55f50bd"),
    '11275': new ObjectId("6416dcb2b09d0a38d55f50be"),
    '11276': new ObjectId("6416dcb2b09d0a38d55f50bf"),
    '11277': new ObjectId("6416dcb2b09d0a38d55f50c0"),
    '11278': new ObjectId("6416dcb2b09d0a38d55f50c1"),
    '11279': new ObjectId("6416dcb2b09d0a38d55f50c2"),
    '11280': new ObjectId("6416dcb2b09d0a38d55f50c3"),
    '11281': new ObjectId("6416dcb2b09d0a38d55f50c4"),
    '11282': new ObjectId("6416dcb2b09d0a38d55f50c5"),
    '11283': new ObjectId("6416dcb2b09d0a38d55f50c6"),
    '11284': new ObjectId("6416dcb2b09d0a38d55f50c7"),
    '11285': new ObjectId("6416dcb2b09d0a38d55f50c8"),
    '11286': new ObjectId("6416dcb2b09d0a38d55f50c9"),
    '11287': new ObjectId("6416dcb2b09d0a38d55f50ca"),
    '11288': new ObjectId("6416dcb2b09d0a38d55f50cb"),
    '11289': new ObjectId("6416dcb2b09d0a38d55f50cc")
  }
}
```

2. Polls the Binance Swap Pool API(s) for swap pool data like the list of coins supported for swaps:

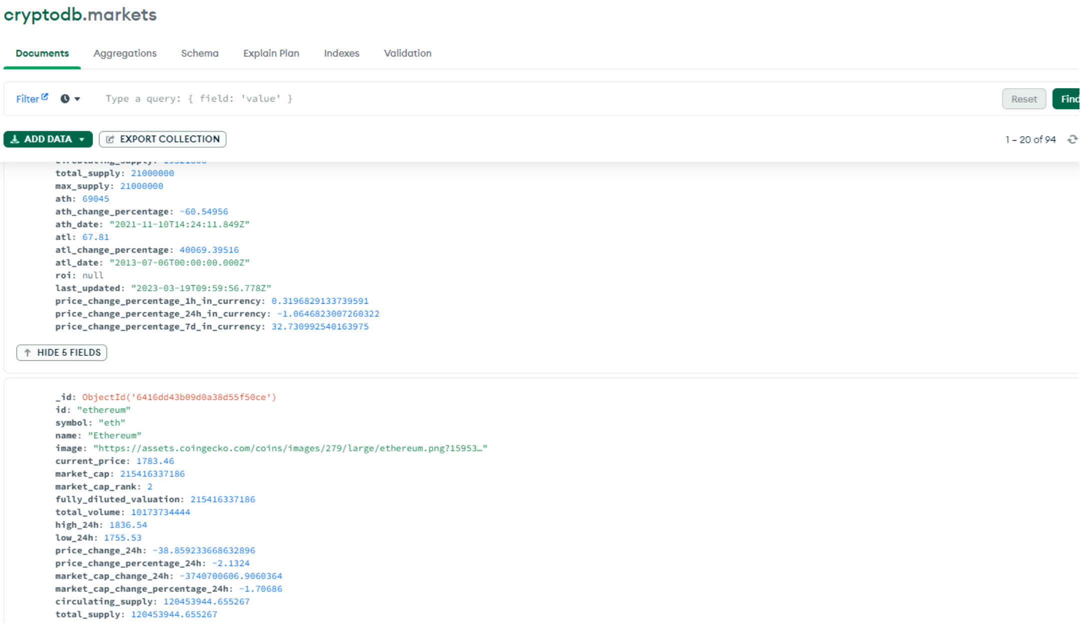


cryptodb.swappools

```
_id: ObjectId('6405b415021a91b582b6b496')
poolId: 18
poolName: "BNB/BTC"
assets: Array
  0: "BNB"
  1: "BTC"

_id: ObjectId('6405b415021a91b582b6b497')
poolId: 19
poolName: "BNB/ETH"
assets: Array

_id: ObjectId('6405b415021a91b582b6b498')
poolId: 25
poolName: "ETH/BETH"
assets: Array
```

```
  'BETH',  'BNB',     'BTC',   'BUSD',  'ETH',    'DAI',     'HOOK',
  'APT',   'OP',      'HFT',   'MATIC', 'DUSK',   'DOGE',    'LINK',
  'AXS',   'MANA',    'USDT',  'THETA', 'CHZ',    'AAVE',    'ADA',
  'FIL',   'BCH',     'LTC',   'FUN',   'NEO',    'AVA',     'SOL',
  'TRX',   'BAKE',    'SHIB',  'CAKE',  'PHA',    'ALICE',   'SXP',
  'ONE',   'DYDX',    'KSM',   'AVAX',  'NEAR',   'ICP',     'PERP',
  'TLM',   'SANTOS',  'DAR',   'ALGO',  'RUNE',   'LIT',     'LOKA',
  'MBOX',  'ILV',     'SAND',  'DOT',   'GALA',   'ALPINE',  'ATA',
  'ATOM',  'GMT',     'APE',   'UNI',   'SUSHI',  'BDOT',    'MULTI',
  'ANT',   'HARD',    'SC',    'IDEX',  'GAL',    'ENJ',     'ELF',
  'TKO',   'PAXG',    'OOKI',  'BSW',   'ETC',    'FTM',     'LDO',
  'KDA',   'IOTX',    'LEVER', 'MDX',   'ARPA',   'MAGIC',   'CRV',
  'GRT',   'FET',     'RPL',   'STG',   'AGIX',   'GMX',     'MKR',
  'SNX',   'LQTY',    'USDC'
]
Connected successfully to MongoDB Server
[
  'beth',  'bnb',     'btc',   'busd',  'eth',    'dai',     'hook',
  'apt',   'op',      'hft',   'matic', 'dusk',   'doge',    'link',
  'axs',   'mana',    'usdt',  'theta', 'chz',    'aave',    'ada',
  'fil',   'bch',     'ltc',   'fun',   'neo',    'ava',     'sol',
  'trx',   'bake',    'shib',  'cake',  'pha',    'alice',   'sxp',
  'one',   'dydx',    'ksm',   'avax',  'near',   'icp',     'perp',
  'tlm',   'santos',  'dar',   'algo',  'rune',   'lit',     'loka',
  'mbox',  'ilv',     'sand',  'dot',   'gala',   'alpine',  'ata',
  'atom',  'gmt',     'ape',   'uni',   'sushi',  'bdot',    'multi',
  'ant',   'hard',    'sc',    'idex',  'gal',    'enj',     'elf',
  'tko',   'paxg',    'ooki',  'bsw',   'etc',    'ftm',     'ldo',
  'kda',   'iotx',    'lever', 'mdx',   'arpa',   'magic',   'crv',
  'grt',   'fet',     'rpl',   'stg',   'agix',   'gmx',     'mkr',
  'snx',   'lqty',    'usdc'
]
```

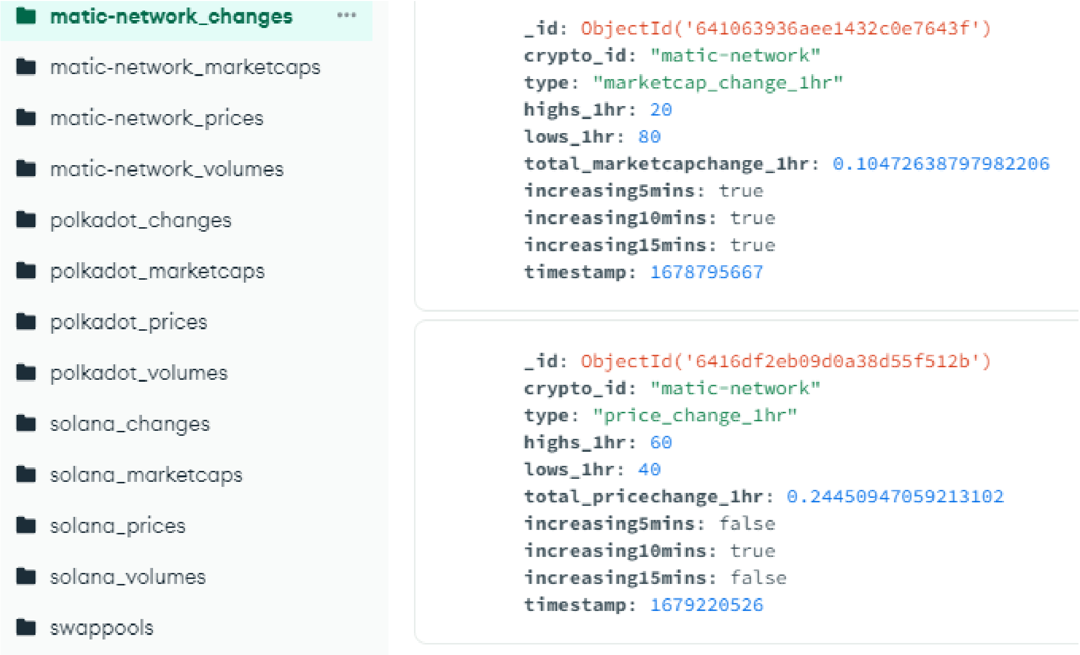
3. Polls the CoinGecko API(s) once again for the granular 5-minute block data (prices, market caps and volumes) of the supported swap coins/tokens:



cryptodb.markets

```
total_supply: 21000000
max_supply: 21000000
ath: 69045
ath_change_percentage: -60.54956
ath_date: "2021-11-10T14:24:11.849Z"
atl: 67.81
atl_change_percentage: 40069.39516
atl_date: "2013-07-06T00:00:00.000Z"
roi: null
last_updated: "2023-03-19T09:59:56.778Z"
price_change_percentage_1h_in_currency: 0.3196829133739591
price_change_percentage_24h_in_currency: -1.0646823007260322
price_change_percentage_7d_in_currency: 32.730992540163975

_id: ObjectId('6416dd43b09d0a38d55f50ce')
id: "ethereum"
symbol: "eth"
name: "Ethereum"
image: "https://assets.coingecko.com/coins/images/279/large/ethereum.png?15953…"
current_price: 1783.46
market_cap: 215416337186
market_cap_rank: 2
fully_diluted_valuation: 215416337186
total_volume: 10173734444
high_24h: 1836.54
low_24h: 1755.53
price_change_24h: -38.859233668632896
price_change_percentage_24h: -2.1324
market_cap_change_24h: -3740700606.9060364
market_cap_change_percentage_24h: -1.70686
circulating_supply: 120453944.655267
total_supply: 120453944.655267
```

4. Calculates 5-minute block changes for each supported coin/token:

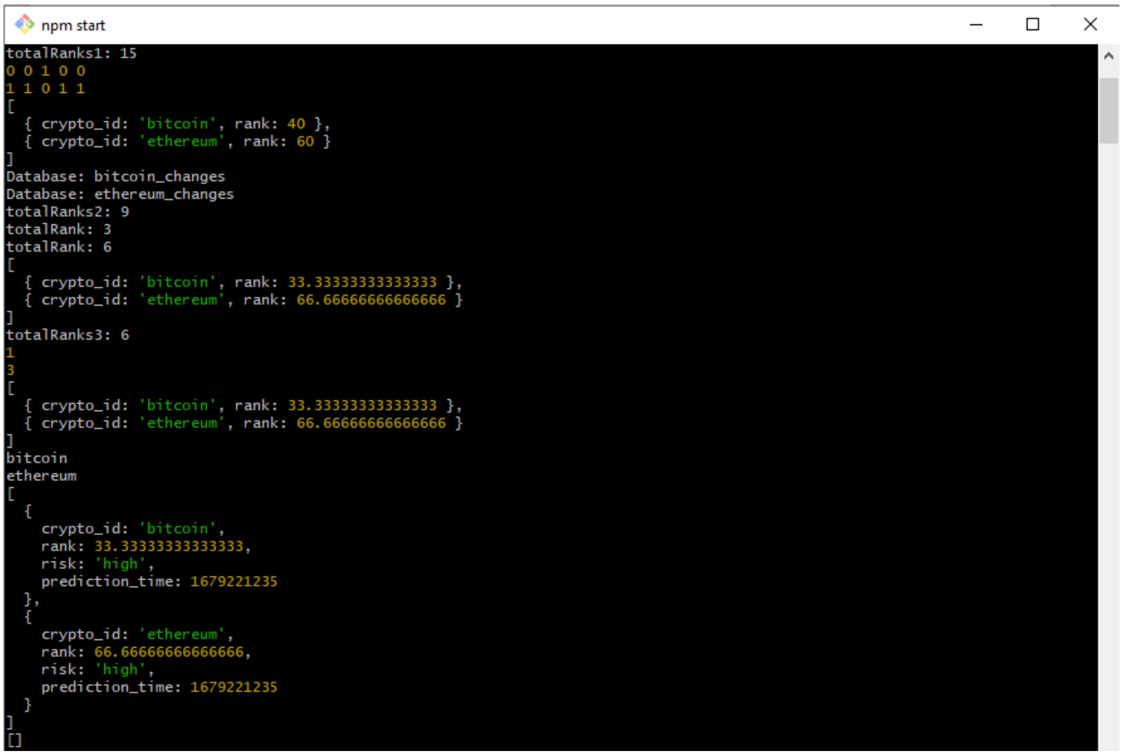


```
matic-network_changes
matic-network_marketcaps
matic-network_prices
matic-network_volumes
polkadot_changes
polkadot_marketcaps
polkadot_prices
polkadot_volumes
solana_changes
solana_marketcaps
solana_prices
solana_volumes
swappools
```

```
_id: ObjectId('641063936aee1432c0e7643f')
crypto_id: "matic-network"
type: "marketcap_change_1hr"
highs_1hr: 20
lows_1hr: 80
total_marketcapchange_1hr: 0.10472638797982206
increasing5mins: true
increasing10mins: true
increasing15mins: true
timestamp: 1678795667
```

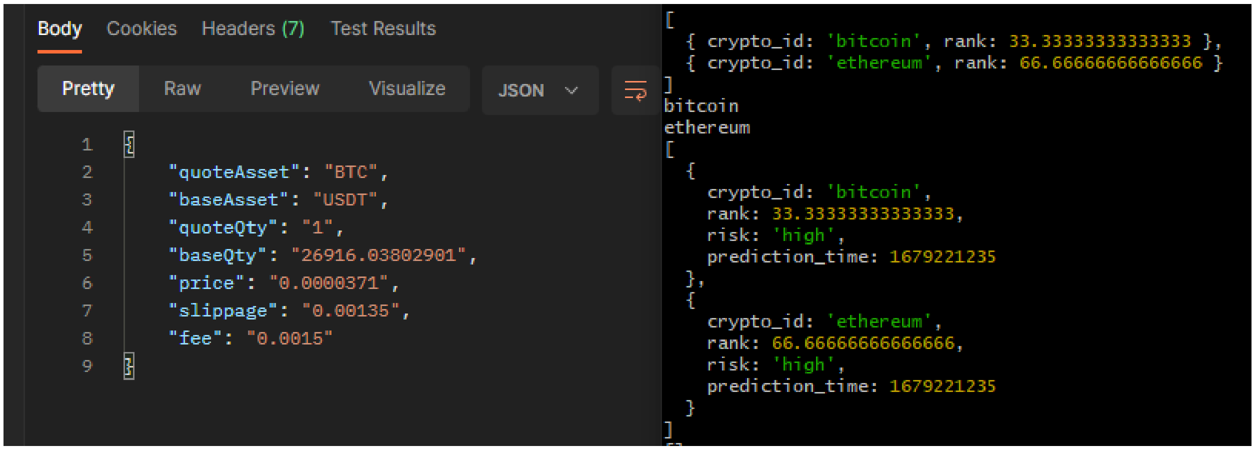
```
_id: ObjectId('6416df2eb09d0a38d55f512b')
crypto_id: "matic-network"
type: "price_change_1hr"
highs_1hr: 60
lows_1hr: 40
total_pricechange_1hr: 0.24450947059213102
increasing5mins: false
increasing10mins: true
increasing15mins: false
timestamp: 1679220526
```

7. Using a proprietary Prediction Model, the bot calculates a rank to risk ratio for each coin:



```
totalRanks1: 15
0 0 1 0 0
1 1 0 1 1
[
  { crypto_id: 'bitcoin', rank: 40 },
  { crypto_id: 'ethereum', rank: 60 }
]
Database: bitcoin_changes
Database: ethereum_changes
totalRanks2: 9
totalRank: 3
totalRank: 6
[
  { crypto_id: 'bitcoin', rank: 33.33333333333333 },
  { crypto_id: 'ethereum', rank: 66.66666666666666 }
]
totalRanks3: 6
1
3
[
  { crypto_id: 'bitcoin', rank: 33.33333333333333 },
  { crypto_id: 'ethereum', rank: 66.66666666666666 }
]
bitcoin
ethereum
[
  {
    crypto_id: 'bitcoin',
    rank: 33.33333333333333,
    risk: 'high',
    prediction_time: 1679221235
  },
  {
    crypto_id: 'ethereum',
    rank: 66.66666666666666,
    risk: 'high',
    prediction_time: 1679221235
  }
]
[]
```

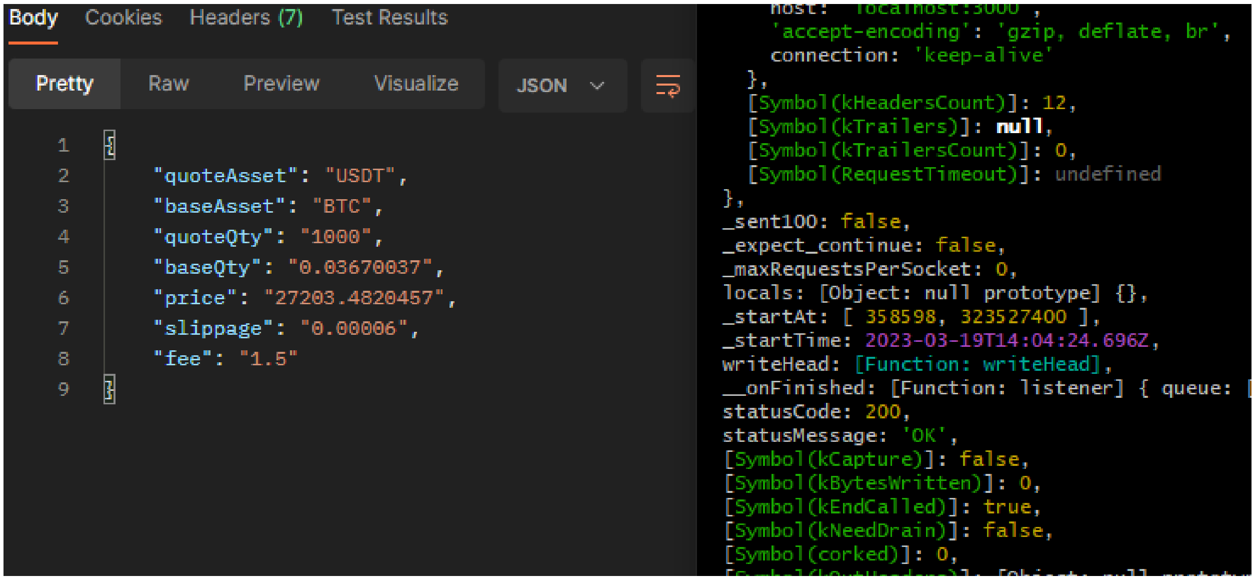
8. Uses the Binance Swap Pool API(s) to make an instantaneous swap from any one stable coin to a non-stable coin:



```
{
    "quoteAsset": "BTC",
    "baseAsset": "USDT",
    "quoteQty": "1",
    "baseQty": "26916.03802901",
    "price": "0.0000371",
    "slippage": "0.00135",
    "fee": "0.0015"
}
```

```
[
  { crypto_id: 'bitcoin', rank: 33.33333333333333 },
  { crypto_id: 'ethereum', rank: 66.66666666666666 }
]
bitcoin
ethereum
[
  {
    crypto_id: 'bitcoin',
    rank: 33.33333333333333,
    risk: 'high',
    prediction_time: 1679221235
  },
  {
    crypto_id: 'ethereum',
    rank: 66.66666666666666,
    risk: 'high',
    prediction_time: 1679221235
  }
]
```

9. Monitors the invested coin for price changes and calculates non-viability of a coin for investment (when the price of the coin is falling). Uses the Binance Swap Pool API(s) again to make an instantaneous swap from the invested non-stable coin to a stable coin:



```
{
    "quoteAsset": "USDT",
    "baseAsset": "BTC",
    "quoteQty": "1000",
    "baseQty": "0.03670037",
    "price": "27203.4820457",
    "slippage": "0.00006",
    "fee": "1.5"
}
```

```
      host: 'localhost:3000',
      'accept-encoding': 'gzip, deflate, br',
      connection: 'keep-alive'
    },
    [Symbol(kHeadersCount)]: 12,
    [Symbol(kTrailers)]: null,
    [Symbol(kTrailersCount)]: 0,
    [Symbol(RequestTimeout)]: undefined
  },
  _sent100: false,
  _expect_continue: false,
  _maxRequestsPerSocket: 0,
  locals: [Object: null prototype] {},
  _startAt: [ 358598, 323527400 ],
  _startTime: 2023-03-19T14:04:24.696Z,
  writeHead: [Function: writeHead],
  __onFinished: [Function: listener] { queue: [
  statusCode: 200,
  statusMessage: 'OK',
  [Symbol(kCapture)]: false,
  [Symbol(kBytesWritten)]: 0,
  [Symbol(kEndCalled)]: true,
  [Symbol(kNeedDrain)]: false,
  [Symbol(corked)]: 0,
  [Symbol(kOutHeaders)]: [Object: null prototyp
```

## Calculation of Return on Investment (ROI/APR)

Average Number of Trades in 24 hours - 100

Maximum Number of Trades (Currently Supported) in 24 hours - 288

*If we consider the probability of the prediction bring correct is 50%, then 50% of the trades executed will be successful. Thus,*

Average Loss % from each trade ** - 0.30%

Net Loss % in 24 hours - (100/2) x 0.30% = 15%

Average Profit % from each trade ** - 0.50%

Net Profit % in 24 hours - (100/2) x 0.50% = 25%

*(IF Average Number of Trades in 24 hours - 100)*

Net PNL % in 24 hours - 25-15% = 10%

**Average PNL % in 1 year (APR) - 10% x 365 = 3,650%**

*(IF Average Number of Trades in 24 hours - 288)*

Net PNL % in 24 hours - 28.8%

Average PNL % in 1 year (APR) - 28.8% x 365 = 10,512%

*These are estimates and actual figures may vary between 500-15,000% APR depending upon how well the market is performing and how the individual coins available in the swap pool are doing in the market.*

*** Excluding Gas Fees*

## Calculation of Fees

Development Fees % - 0.05% (Paid to Developer)

LUNC Burn % - 0.05% (LUNC Burnt)

**Net Fees % per transaction - 0.1%**

Average Number of Trades in 24 hours - 100

Average Value of each trade - $100

Net Development Fees in USD in 24 hours per user - $100 x 0.05% x 100 = $5

Net LUNC Burns in USD in 24 hours per user - $100 x 0.05% x 100 = $5

*If we consider that only 1000 users are using the bot every day,*

**Volume of trade per day** - $100 x 1000 x 100 = **$10,000,000** (total for all coins, not just LUNC)

**Net LUNC Burns** in USD in 24 hours for 1000 users - $5 x 1000

= $5,000 or ~**38,461,538 per day** @ 0.00013 USD

*These are estimates and actual figures may vary, though the figure above is underquoted and fairly close to expected returns/burns. The actual returns/burns may be much higher than this.*

## What is the current state of development and who are the developers working on it? Is the code open source?

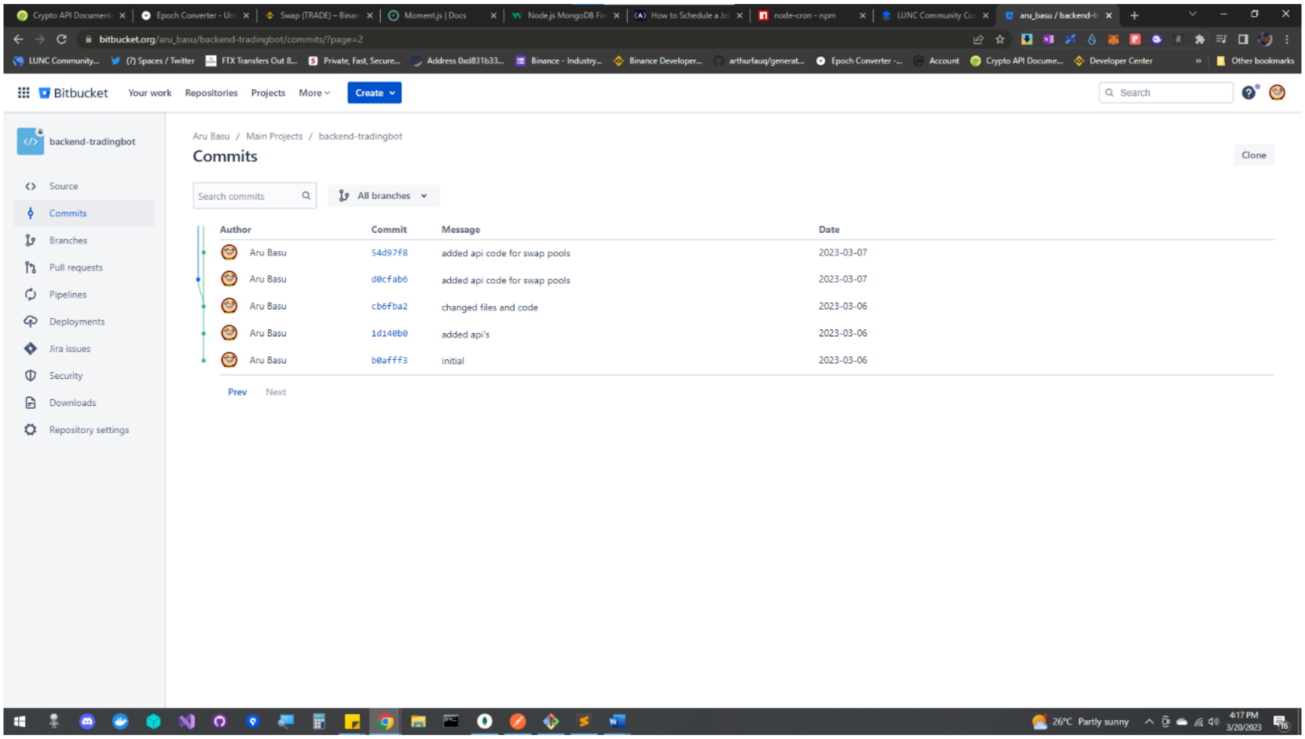
### Developers

I, Arunaday Basu, am the sole developer on this project currently, but the investment required for this project includes 1-2 developers apart from me (who will be required if this proposal is passed). I am primarily a full-stack JavaScript developer and MERN (MongoDB, Express, React, Node) is the stack that I generally work on. I have almost 13 years of development experience in various languages, stacks and CMS. I have worked for various companies including blockchain/cryptocurrency startups. I am professionally a business owner and I have a Bachelor of Engineering degree in Computer Science and Technology. I am a resident of India, and apart from coding, I am passionate about trading on the cryptocurrency market, mining, history and art. A longer description of myself is provided in the other proposal I have submitted for discussion.

### Current State of Development

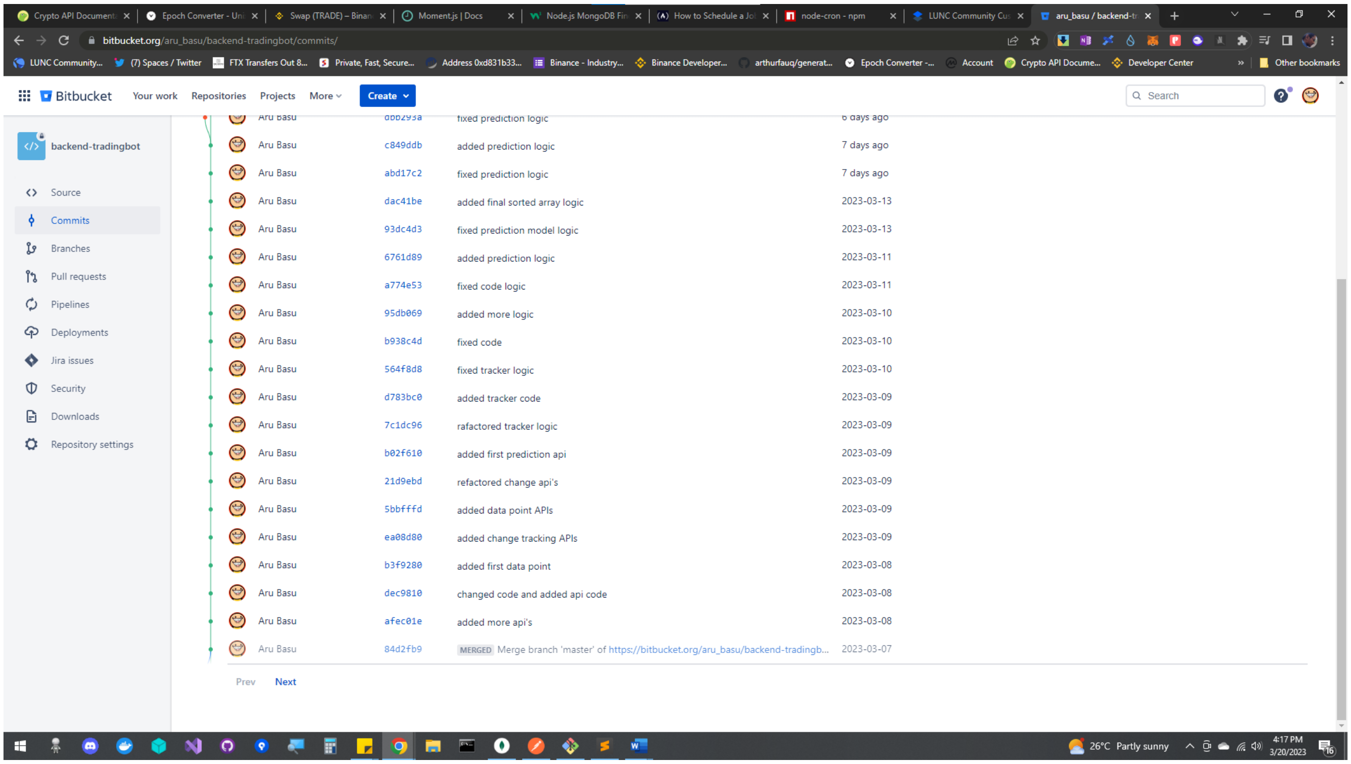
I have started writing code this month on the 6th. In the last few days, I have almost finished writing most of the APIs including fetching data from different trackers, saving data in a database, calculating investment viability and executing trades.

Since I do not push to Github generally for work (I do have a public profile there), I am providing screenshots below of the repo in my private BitBucket account:

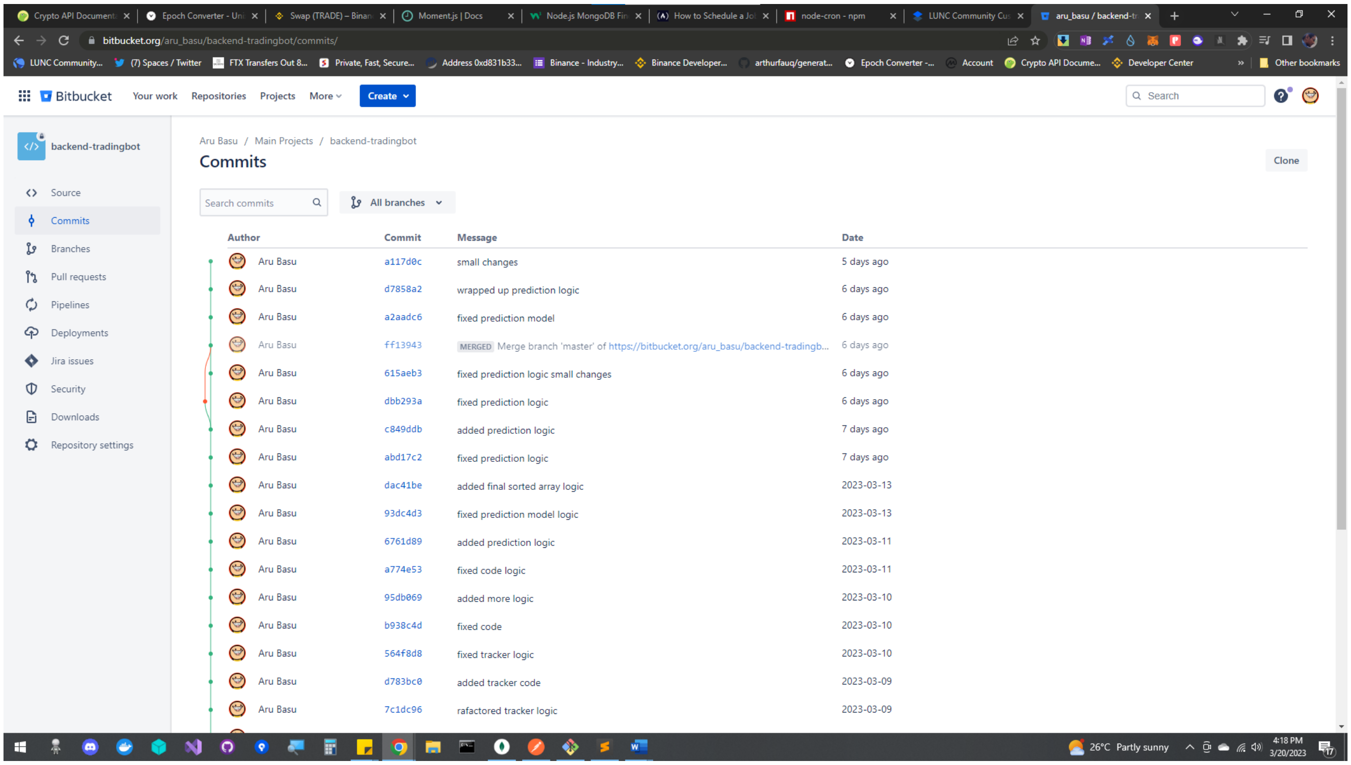


Commits

| Author | Commit | Message | Date |
|---|---|---|---|
| Aru Basu | 54d97f8 | added api code for swap pools | 2023-03-07 |
| Aru Basu | d0cfab6 | added api code for swap pools | 2023-03-07 |
| Aru Basu | cb6fba2 | changed files and code | 2023-03-06 |
| Aru Basu | 1d140b0 | added api's | 2023-03-06 |
| Aru Basu | b0afff3 | initial | 2023-03-06 |



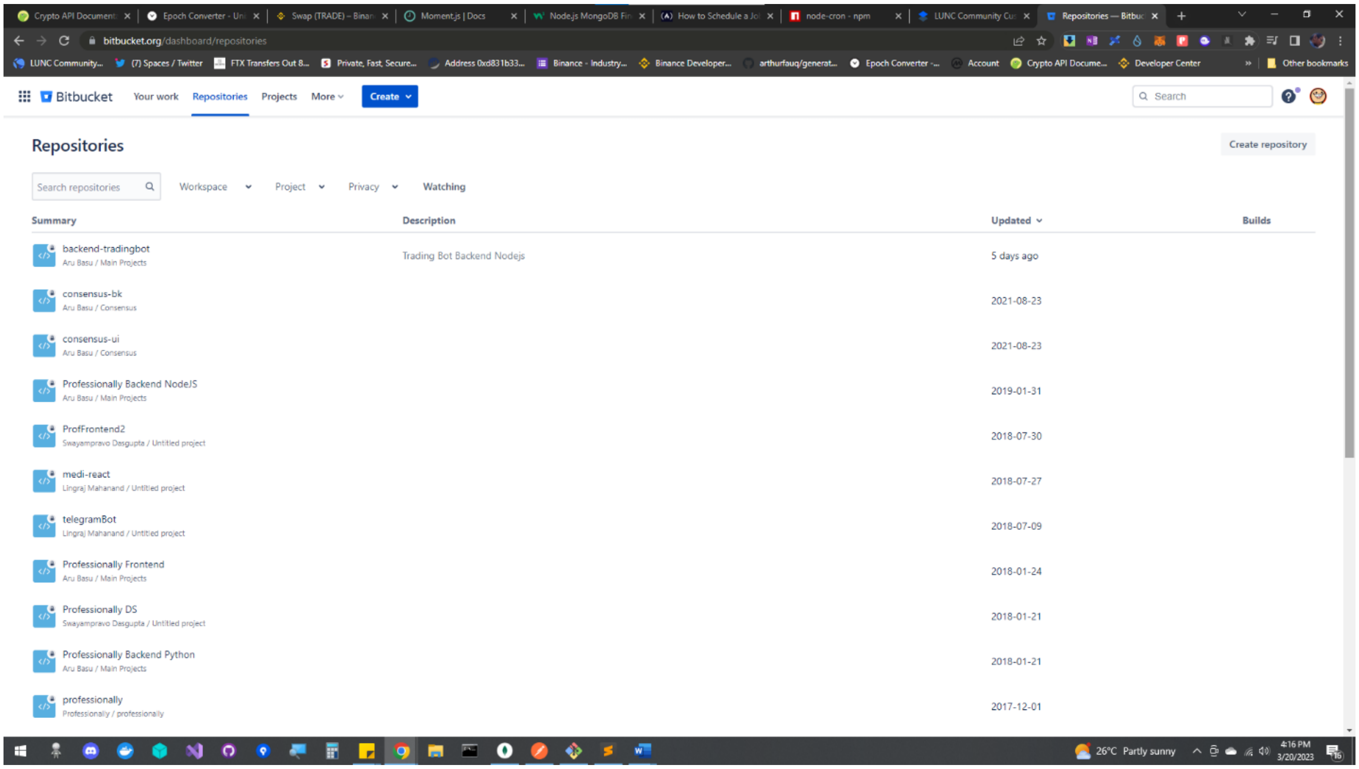
| Author | Commit | Message | Date |
|---|---|---|---|
| Aru Basu | dbb293a | fixed prediction logic | 6 days ago |
| Aru Basu | c849ddb | added prediction logic | 7 days ago |
| Aru Basu | abd17c2 | fixed prediction logic | 7 days ago |
| Aru Basu | dac41be | added final sorted array logic | 2023-03-13 |
| Aru Basu | 93dc4d3 | fixed prediction model logic | 2023-03-13 |
| Aru Basu | 6761d89 | added prediction logic | 2023-03-11 |
| Aru Basu | a774e53 | fixed code logic | 2023-03-11 |
| Aru Basu | 95db069 | added more logic | 2023-03-10 |
| Aru Basu | b938c4d | fixed code | 2023-03-10 |
| Aru Basu | 564f8d8 | fixed tracker logic | 2023-03-10 |
| Aru Basu | d783bc0 | added tracker code | 2023-03-09 |
| Aru Basu | 7c1dc96 | rafactored tracker logic | 2023-03-09 |
| Aru Basu | b02f610 | added first prediction api | 2023-03-09 |
| Aru Basu | 21d9ebd | refactored change api's | 2023-03-09 |
| Aru Basu | 5bbfffd | added data point APIs | 2023-03-09 |
| Aru Basu | ea08d80 | added change tracking APIs | 2023-03-09 |
| Aru Basu | b3f9280 | added first data point | 2023-03-08 |
| Aru Basu | dec9810 | changed code and added api code | 2023-03-08 |
| Aru Basu | afec01e | added more api's | 2023-03-08 |
| Aru Basu | 84d2fb9 | MERGED Merge branch 'master' of https://bitbucket.org/aru_basu/backend-tradingb... | 2023-03-07 |



Aru Basu / Main Projects / backend-tradingbot

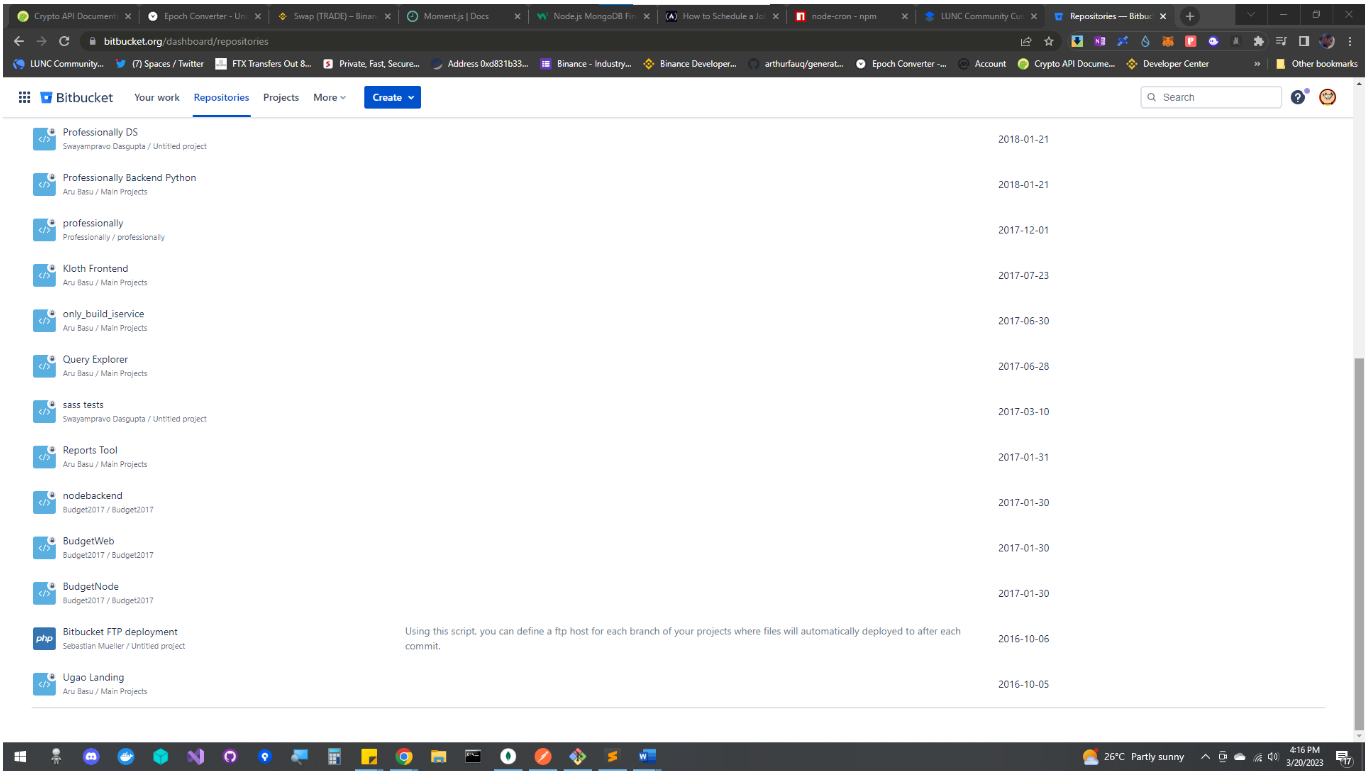
## Commits

| Author | Commit | Message | Date |
|---|---|---|---|
| Aru Basu | a117d0c | small changes | 5 days ago |
| Aru Basu | d7858a2 | wrapped up prediction logic | 6 days ago |
| Aru Basu | a2aadc6 | fixed prediction model | 6 days ago |
| Aru Basu | ff13943 | MERGED Merge branch 'master' of https://bitbucket.org/aru_basu/backend-tradingb... | 6 days ago |
| Aru Basu | 615aeb3 | fixed prediction logic small changes | 6 days ago |
| Aru Basu | dbb293a | fixed prediction logic | 6 days ago |
| Aru Basu | c849ddb | added prediction logic | 7 days ago |
| Aru Basu | abd17c2 | fixed prediction logic | 7 days ago |
| Aru Basu | dac41be | added final sorted array logic | 2023-03-13 |
| Aru Basu | 93dc4d3 | fixed prediction model logic | 2023-03-13 |
| Aru Basu | 6761d89 | added prediction logic | 2023-03-11 |
| Aru Basu | a774e53 | fixed code logic | 2023-03-11 |
| Aru Basu | 95db069 | added more logic | 2023-03-10 |
| Aru Basu | b938c4d | fixed code | 2023-03-10 |
| Aru Basu | 564f8d8 | fixed tracker logic | 2023-03-10 |
| Aru Basu | d783bc0 | added tracker code | 2023-03-09 |
| Aru Basu | 7c1dc96 | rafactored tracker logic | 2023-03-09 |



## Repositories

| Summary | Description | Updated | Builds |
|---|---|---|---|
| backend-tradingbot<br>Aru Basu / Main Projects | Trading Bot Backend Nodejs | 5 days ago | |
| consensus-bk<br>Aru Basu / Consensus | | 2021-08-23 | |
| consensus-ui<br>Aru Basu / Consensus | | 2021-08-23 | |
| Professionally Backend NodeJS<br>Aru Basu / Main Projects | | 2019-01-31 | |
| ProfFrontend2<br>Swayampravo Dasgupta / Untitled project | | 2018-07-30 | |
| medi-react<br>Lingraj Mahanand / Untitled project | | 2018-07-27 | |
| telegramBot<br>Lingraj Mahanand / Untitled project | | 2018-07-09 | |
| Professionally Frontend<br>Aru Basu / Main Projects | | 2018-01-24 | |
| Professionally DS<br>Swayampravo Dasgupta / Untitled project | | 2018-01-21 | |
| Professionally Backend Python<br>Aru Basu / Main Projects | | 2018-01-21 | |
| professionally<br>Professionally / professionally | | 2017-12-01 | |



| | | |
|---|---|---|
| Professionally DS<br>Swayampravo Dasgupta / Untitled project | | 2018-01-21 |
| Professionally Backend Python<br>Aru Basu / Main Projects | | 2018-01-21 |
| professionally<br>Professionally / professionally | | 2017-12-01 |
| Kloth Frontend<br>Aru Basu / Main Projects | | 2017-07-23 |
| only_build_iservice<br>Aru Basu / Main Projects | | 2017-06-30 |
| Query Explorer<br>Aru Basu / Main Projects | | 2017-06-28 |
| sass tests<br>Swayampravo Dasgupta / Untitled project | | 2017-03-10 |
| Reports Tool<br>Aru Basu / Main Projects | | 2017-01-31 |
| nodebackend<br>Budget2017 / Budget2017 | | 2017-01-30 |
| BudgetWeb<br>Budget2017 / Budget2017 | | 2017-01-30 |
| BudgetNode<br>Budget2017 / Budget2017 | | 2017-01-30 |
| Bitbucket FTP deployment<br>Sebastian Mueller / Untitled project | Using this script, you can define a ftp host for each branch of your projects where files will automatically deployed to after each commit. | 2016-10-06 |
| Ugao Landing<br>Aru Basu / Main Projects | | 2016-10-05 |

*This is a view of the repos I already have in that Bitbucket account. I am pushing code there for 7 years now.*

The next few developments planned for the Auto Spot Trading Bot are:

1. UI using which users will be able to login and access the features

2. Live crypto prices ticker with predictions that show how the bot works

3. Improvement of the prediction model

There is, of course, a lot of code to be written including making the prediction model better and accurate to the second, but I require investment for this since I will be working full-time on it until the objectives mentioned above are achieved, including the mentioned future plans.

## Proprietary Code

It is not only difficult to maintain this project as an open-source project, it is equally difficult to provide information on it transparently after what has happened with the casino and how the code was probably copied by another entity to create a completely different project.

If that happens with this project, not only is that a danger to the security and safety of the automation system and its users, I will not have a representative, company or even individual to pursue for legal and financial damages.

I will not be able to publish this project as an open-source project and the code will completely be owned by me, Arunaday Basu. I will, however, sign a 5 years maintenance contract with the community, during which time all development, including looking after bugs and fixes are my responsibility. I will sign and send this contract to the Terra Grants Foundation physically (in mail) if this proposal is passed.

Within 5 years, I plan to get investment to maintain the dApp further. If that isn't possible for any unavoidable circumstance, I will duly transfer the code to another competent authority representing the Terra Luna Classic community (upon passing a proposal appointing the competent authority as a representative of the community).

## What is the investment/funding required for the project (from the Terra Luna Classic Community Pool)?

Total Investment Required for Development & 5 years Maintenance - **$100,000 USD in LUNC**

(Which is approximately ~769,230,769 LUNC @ 0.00013 USD/LUNC)

Total Time Required for Development - **2-3 months for Beta**, and 1-2 months more to go Live

(4-5 months in total)

The investment required is high, even compared to my other proposal. This is because I need to devote full-time into work that will require me relinquishing my current responsibilities completely. I will probably need to hire and manage a couple of developers along with myself working full-time on coding. So, this is not work that can be done part-time.

The investment required will be used for the following:

1. Cloud server on AWS or GCP for 5 years
2. Paid API plans because the free plans do not support too many transactions
3. Remuneration for developers including ML & AI developers
4. My remuneration for 4-5 months
5. Marketing, funding, and other activities

*Declaration: I hereby declare that the details furnished above are true and correct to the best of my knowledge and belief and I undertake to inform the community of any changes therein, immediately. In case any of the above information is found to be false or untrue or misleading or misrepresenting, I am aware that I may be held liable for it.*

*** A PDF version of this proposal is attached for users who wish to read it offline*