

“CRYPTOLYSER”

A

Major Project

Synopsis

On

Submitted in partial fulfillment for the award

of Bachelor of Technology in

Computer Science & Engineering

Submitted to

RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA

BHOPAL (M.P.)



Submitted By

Dheeraj Kumar - 0131CS181025

Harsh Jain – 0131CS181028

Lekhraj Meena – 0131CS181034

Mahesh Raj Sharma – 0131CS181035

Under the Guidance of

Prof Menali Paul

JAI NARAIN COLLEGE OF TECHNOLOGY, BHOPAL (M.P.)

JAI NARAIN COLLEGE OF TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



CERTIFICATE

This is to certify that Minor project synopsis Project entitled “**Cryptolyser**” submitted to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.) by “**Harsh Jain**” for partial fulfillment for the award of the degree of the Bachelor of Technology in Computer Science & Engineering.

Dheeraj Kumar - 0131CS181025

Harsh Jain– 0131CS181028

Lekhraj Meena – 0131CS181034

Mahesh Raj Sharma – 0131CS181035

Prof. Nikhil Pateria

HOD (CSE)

JNCT, Bhopal (M.P.)

JAI NARAIN COLLEGE OF TECHNOLOGY, BHOPAL
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



APPROVAL CERTIFICATE

This is hereby certified that the Major Project synopsis entitled “**Cryptolyser**” being submitted by “**Harsh Jain 0131CS181028**” to the RGPV, Bhopal is a genuine work performed by him.

Internal Examiner

External Examiner

Date:

Date:

ACKNOWLEDGEMENT

This is one of the best moments of my B.Tech. program to publicly acknowledge those who have contributed in many different ways to make my success a part of their own. The completion of the Major Project depends upon the co-operation, coordination and combined effects of several resources of knowledge energy.

I heartily thanks to Prof **Menali Paul Mam**, who have supported in major project, faculties of Department of Computer Science & Engineering, for accepting me to work under their Valuable Guidance, Closely Supervised this work over the past few months and offering many innovative ideas and helpful suggestions, which led to the successful completion of this dissertation work.

I am especially thankful **Dr. Netra Pal Singh**, Principal, JNCT, and Bhopal for his kind co-operation and rendering me all possible facilities.

I am thankful to all staff members of the CSE department and my friends for their timely help co-operation and suggestion during my project work. Lastly but not the least, I must express thanks to my family, without their moral support it was impossible for me to complete this major project work.

Harsh Jain

.....

Table of Contents	Page No
Abstract	
Chapter 1 Introduction	
1.1 Objective	
1.2 Scope	
1.3 Purpose	
1.4 Problem Statement	
Chapter 2 Design	
2.1 E-R Diagram	
2.2 Data Flow Diagram	
2.3 Class Diagram	
2.4 Sequence Diagram	
2.5 Flow Chart Diagram, etc.	
Chapter 3 Implementation Requirements	
3.1 Front-End	
3.2 Back-End	
Chapter 4 Lay-out	
4.1 Snapshot (if any)	
4.2 Coding (if any)	
4.3 Database Tables (if any)	
Chapter 5 Application	
5.1 Advantage(s)	
5.2 Disadvantage(s)	
5.3 Application(s)	
Chapter 6 Conclusion and Future Work	
6.1 Conclusion	
6.2 Future Work	
References	

List of Figure

Fig No	Figure Name	Page No
Fig 1	E-R diagram	15
Fig 2	Sequence diagram	20
Fig 3	GUI snapshots	26
Fig 4	Program codes snapshots	31

1.INTRODUCTION

1.1 OBJECTIVE:

Cryptocurrency has taken the world by storm, with nearly 7,000 cryptocurrencies currently listed on CoinMarketCap. This has sparked the interest of beginner investors to join the industry. Being that it's a popular investment today, especially among younger investors, it's tempting to jump into the crypto world without sufficient knowledge due to the fear of missing out.

Investing in cryptocurrency is a risky venture under any circumstances, and it's especially so if you have no idea how to research and analyze cryptos. But, with the right crypto research tools, you can gather the information you need to make an informed investment decision.

For this reason, a tool is needed to analysis the market value of currencies and to analyse the real time profits and losses from the respective currencies.

1.2 SCOPE :

Designing the User Interface:

User interface consist of GUI components of Project.

Designing it with a user-friendly approach to sustain a better application.

Preparing the codebase:

Preparing for applied algorithms, abstracts, relations and documents. Designing the base of whole structure to ensure the integrity of different modules.

Understanding the web scrapping:

Web scrapping is essential part to get data for application. Understanding and applying it with application environment.

Building user interface:

Developing the Graphical User Interface and integrating it with backend algorithm.

Developing the Codebase and Documentation:

Developing the Code and preparing the documents. Verifying the application by testing and matching it by given requirement.

Testing By Use Cases:

Preparing for Test cases, Use cases and testing as per requirements.

1.3 PURPOSE:

A cryptocurrency, crypto-currency, or crypto is a collection of binary data which is designed to work as a medium of exchange. Individual coin ownership records are stored in a ledger, which is a computerized database using strong cryptography to secure transaction records, to control the creation of additional coins, and to verify the transfer of coin ownership.

Cryptolyser is a simulation tool and analytic framework which allows user to simulate the cryptocurrencies with their real value in real time.

The purpose of Cryptolyser is to provide a free, hassle-less and user friendly simulation to build experience for crypto trading and to understand fluctuations in them with real time.

1.4 PROBLEM STATEMENT:

Cryptocurrency investing can be an adventure. Once you've set up an account with a cryptocurrency exchange and figured out how (and where) you want to store your crypto assets, you'll need to keep on top of your investments. And since the crypto industry is relatively young, some of the services are not as user friendly hope.

One such service is tracking the value of crypto – asset in real time with accurate market price and in ease way. As the Cryptocurrency market is unstable and changes very frequently (almost every second), a graphical tool is required to keep track of it, and conserve profit and losses of respective currency digitally.

Cryptolyser handles this service as a real time service and count out transaction values currently.

2. DESIGN

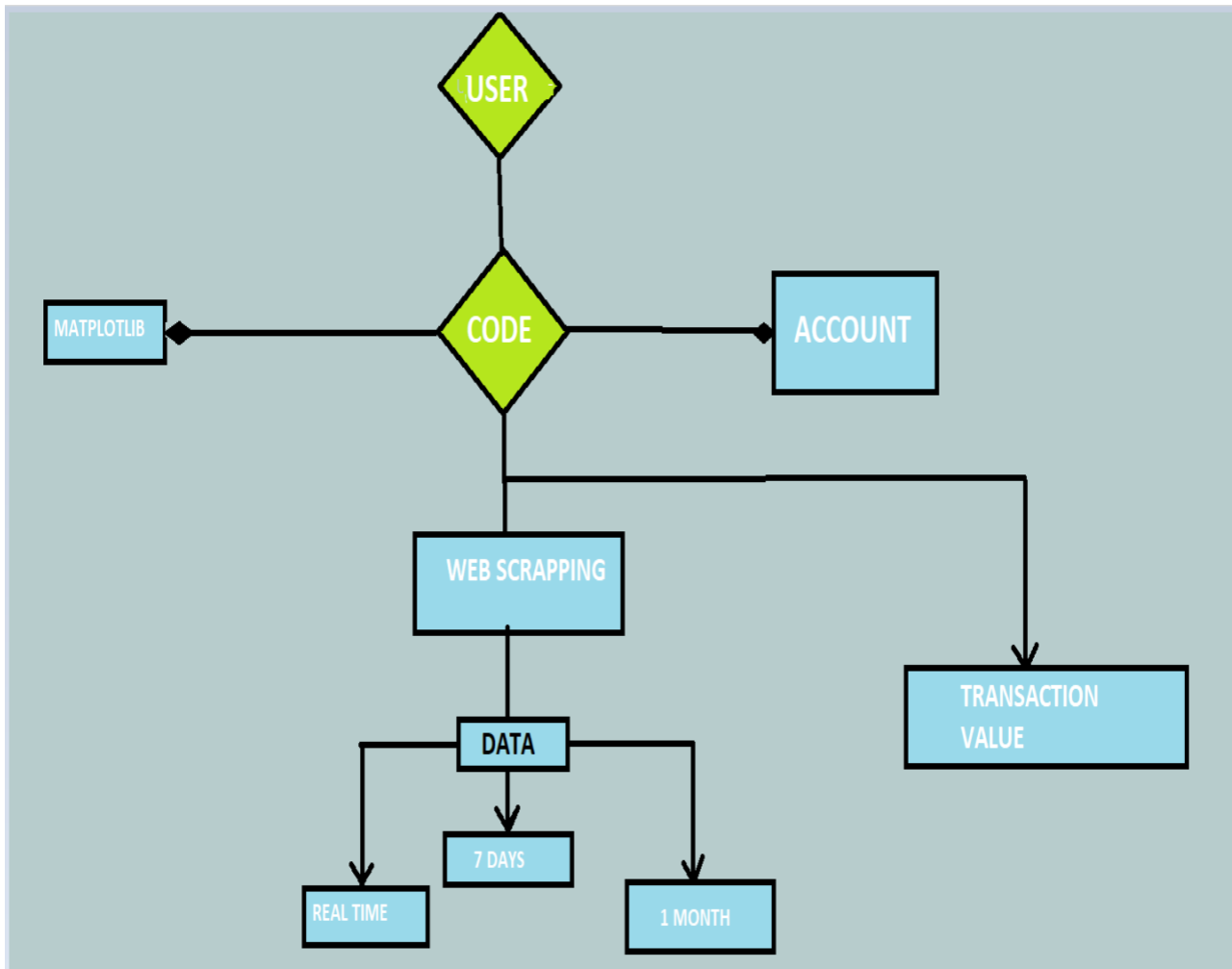
2.1 E-R DIAGRAM:

An entity–relationship model (or ER model) describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between entities (instances of those entity types).

In software engineering, an ER model is commonly formed to represent things a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model, that defines a data or information structure which can be implemented in a database, typically a relational database.

ER diagrams are related to data structure diagrams (DSDs), which focus on the relationships of elements within entities instead of relationships between entities themselves. ER diagrams also are often used in conjunction with data flow diagrams (DFDs), which map out the flow of information for processes or systems.

Cryptolyser build on following E-R relation:

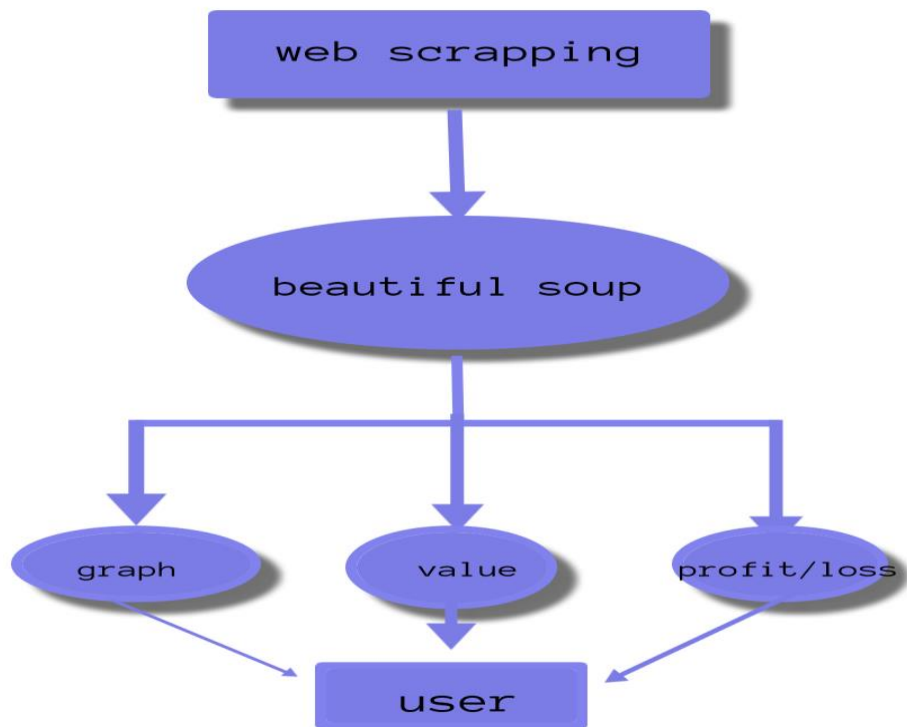


2.2 DATA FLOW DIGRAM:

A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow — there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.

In Crptolyser, the main source of data to be processed is internet. Data is taken from a crypto currency website and treated to give desired output.

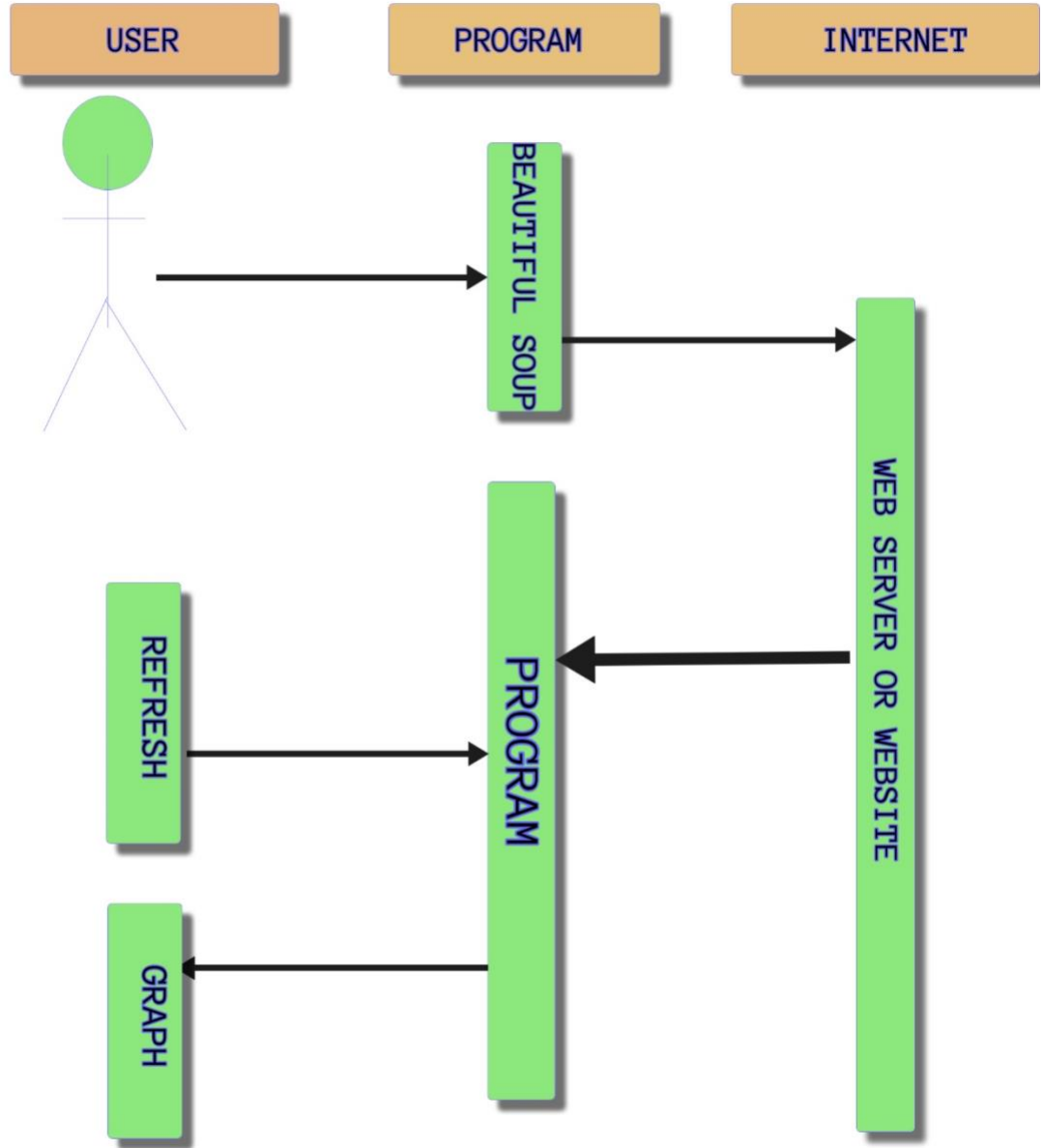
The main model for data flow in Cryptolyser can be stated as follows:



2.3 SEQUENCE DIAGRAM :

A sequence diagram or system sequence diagram (SSD) shows object interactions arranged in time sequence in the field of software engineering. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

Cryptolyser brings upon following sequence diagram.



3. IMPLEMENTATION REQUIREMENTS:

FRONT – END:

Cryptolyser is an open GUI based standalone desktop application program, which executes using a python shell environment. The program uses python graphical user interface building and developing libraries and visualization packages to show users its proceed outputs and processed results.

The main front – end of Cryptolyser are:

1. Tkinter library :

The tkinter package (“Tk interface”) is the standard Python interface to the Tcl/Tk GUI toolkit. Both Tk and tkinter are available on most Unix platforms, including macOS, as well as on Windows systems. The tkinter package (“Tk interface”) is the standard Python interface to the Tcl/Tk GUI toolkit. Both Tk and tkinter are available on most Unix platforms, including macOS, as well as on Windows systems.

2. Matplotlib library package:

Matplotlib use a rich array of third-party packages built on. Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

- Create publication quality plots.
- Make interactive figures that can zoom, pan, update.
- Customize visual style and layout.
- Export to many file formats .
- Embed in JupyterLab and Graphical User Interfaces.

Matplotlib is used as a plotting tool for plotting the real time values and to determine the scope of values over a range of time period.

BACK – END

The back – end of Cryptolyser is a Python based system, which runs on the python (Version 3) and has a stable tested and reliable code.

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is used as an intermediary language and application uses various libraries and subsystems of python to executes tasks.

Following libraries are used as tools for developing Cryptolyser:

Beautiful Soup:

Beautiful Soup is a Python package for parsing HTML and XML documents (including having malformed markup, i.e. non-closed tags, so named after tag soup). It creates a parse tree for parsed pages that can be used to extract data from HTML, which is useful for web scraping.

Cryptolyser extracts web data from the HTML/XML structures present in the webpages of several public crypto – currencies websites and convert them to the python objects and structures. These objects then used by code to process results.

Selenium :

Selenium is an open-source umbrella project for a range of tools and libraries aimed at supporting web browser automation. Selenium provides a playback tool for authoring functional tests without the need to learn a test scripting language (Selenium IDE).

Cryptolyser uses selenium libraries provided by python package manager to run the automation with the help of Chrome browser (Headless mode). Chrome browser searches and opens webpages from which code requires to extract and scrap data.

Multiprocessing :

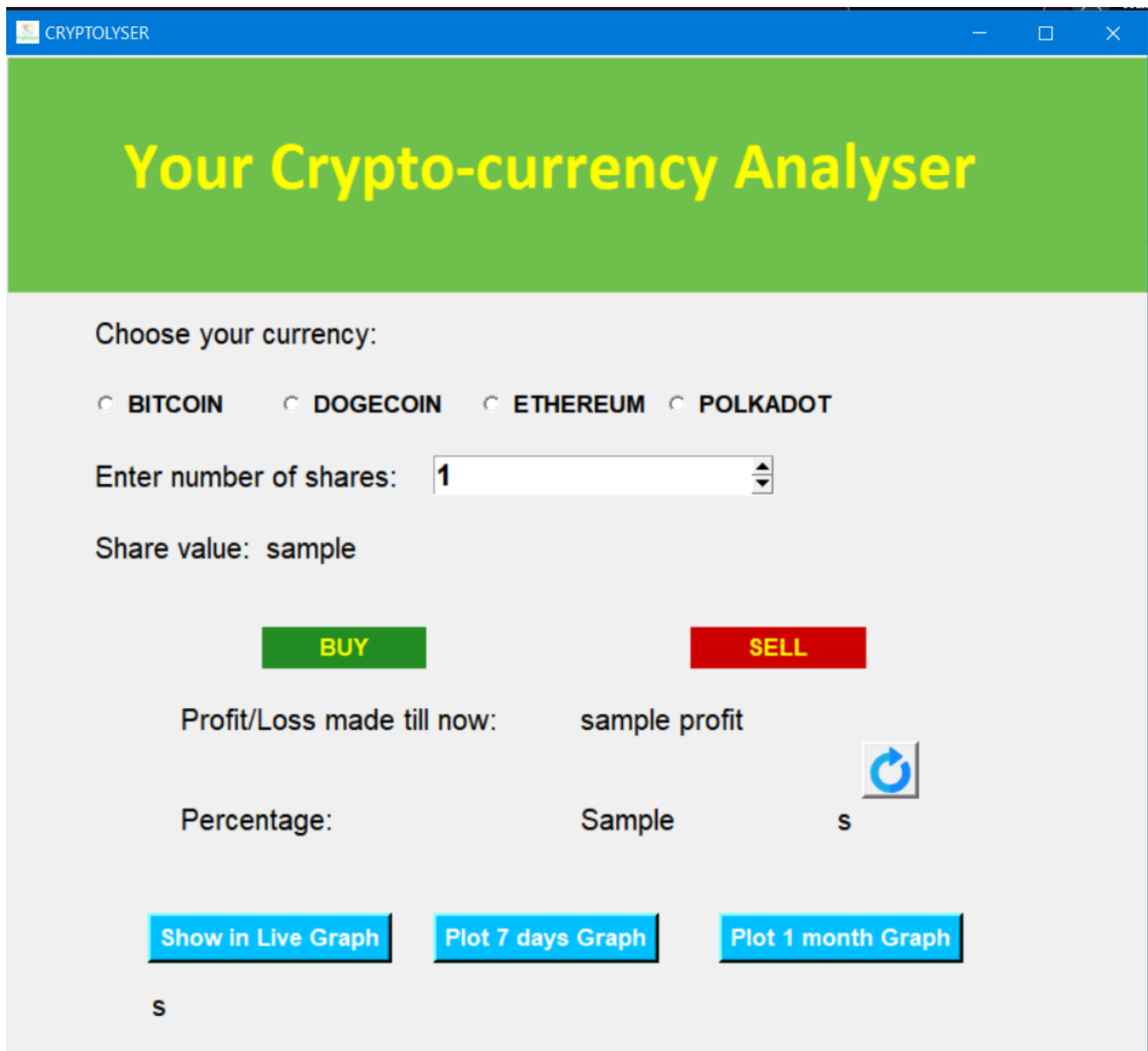
Multiprocessing is the use of two or more central processing units (CPUs) within a single computer system. The term also refers to the ability of a system to support more than one processor or the ability to allocate tasks between them. There are many variations on this basic theme, and the definition of multiprocessing can vary with context, mostly as a function of how CPUs are defined (multiple cores on one die, multiple dies in one package, multiple packages in one system unit, etc.).

In Cryptolyser, it is required to run the GUI and plotting graphs simultaneously and without being dependent on each other. That is, the both process requires to be mutually exclusive of each other, while being sharing the same code at backend. So multiprocessing library from the python package provider imported and used to run the Matplotlib graph and Tkinter as two separate processes.

4.LAYOUTS

○ Some snapouts of the project is shown here as follows:

1. The Main window of Cryptolyser.



CRYPTOLYSER

Your Crypto-currency Analyser

Choose your currency:


☐ BITCOIN ☐ DOGECOIN ☐ ETHEREUM ☐ POLKADOT

Enter number of shares:

Share value: sample

BUY **SELL**

Profit/Loss made till now: sample profit

Percentage: Sample 

s

Show in Live Graph **Plot 7 days Graph** **Plot 1 month Graph**

s

2. Cryptolyser calculating profit and loss after buying shares:

CRYPTOLYSER

Your Crypto-currency Analyser

Choose your currency:

☐ BITCOIN

☐ BINANCE

☐ ETHEREUM

☐ POLKADOT

Enter number of shares:

1

Share value:

54,679.0

BUY

SELL

Profit/Loss made till now:

-26.90000

Percentage:


-0.05

Show in Live Graph

Plot 7 days Graph

Plot 1 month Graph

Item Buy: BITCOIN

 CRYPTOLYSER

Your Crypto-currency Analyser

Choose your currency:

☐ BITCOIN ☐ BINANCE ☐ ETHEREUM ☐ POLKADOT


Enter number of shares:

Share value: 54,679.0

BUY

SELL

Profit/Loss made till now: 12.30000

Percentage:  0.02

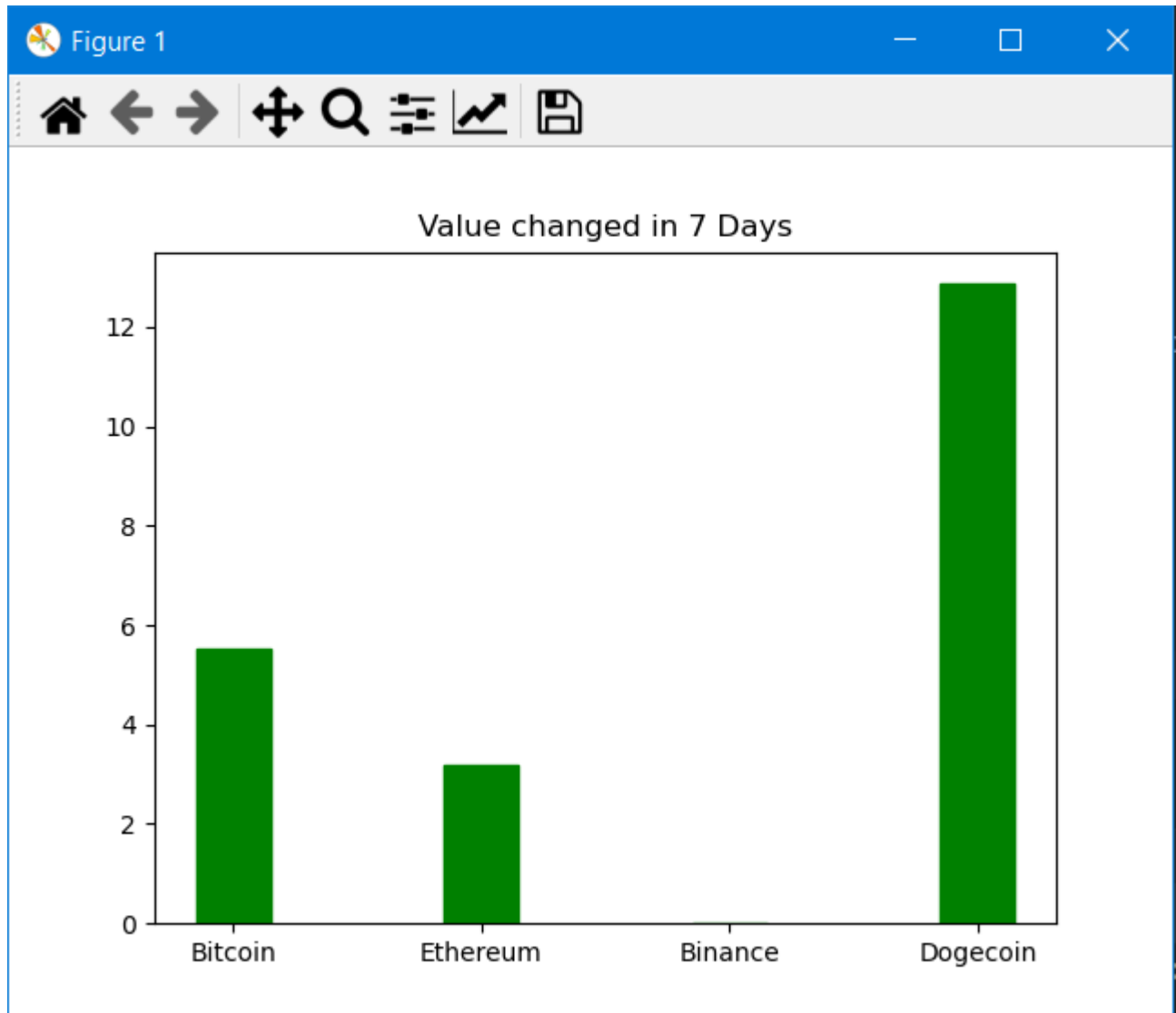
Show in Live Graph

Plot 7 days Graph

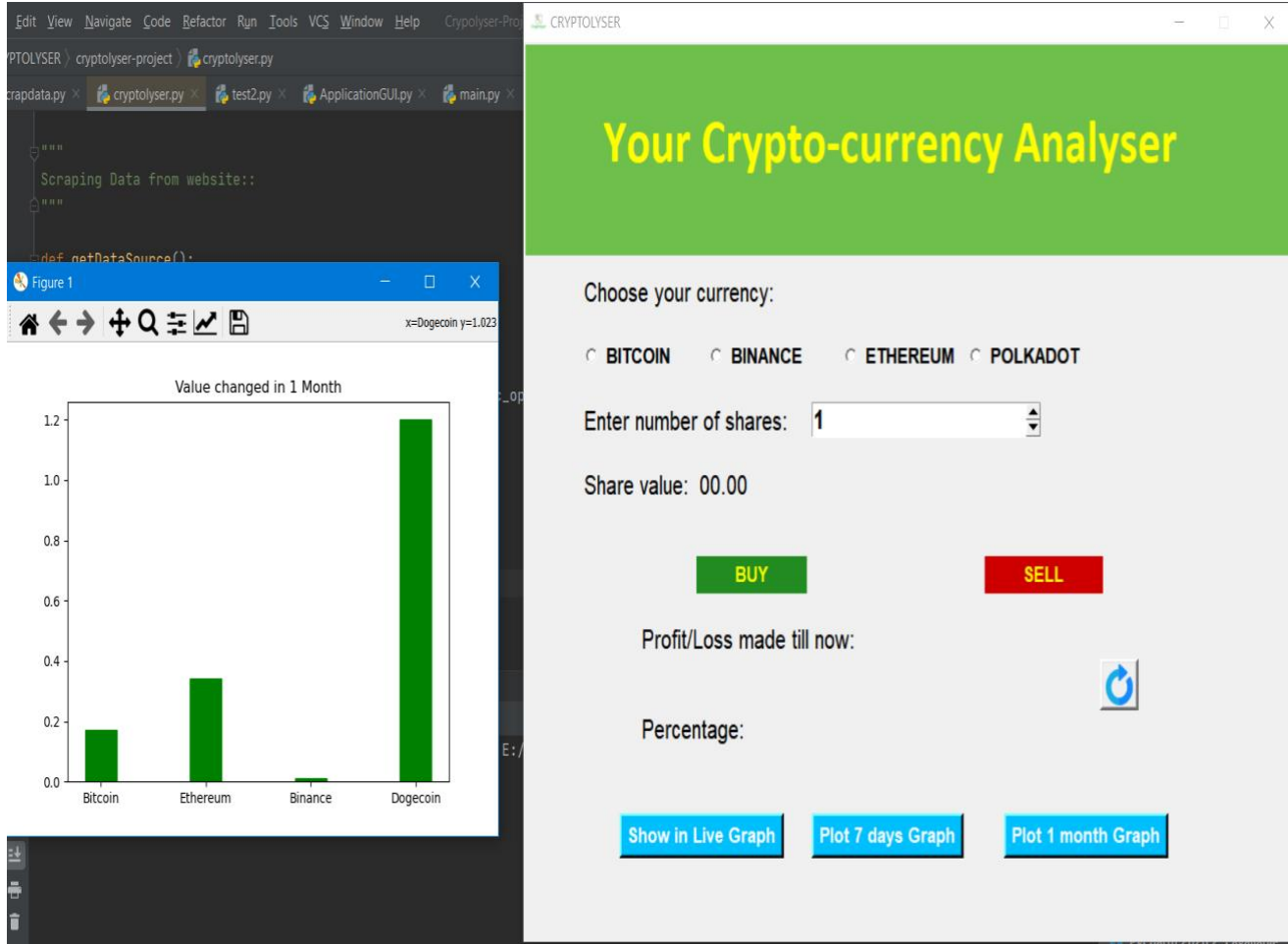
Plot 1 month Graph

Item Buy: BITCOIN

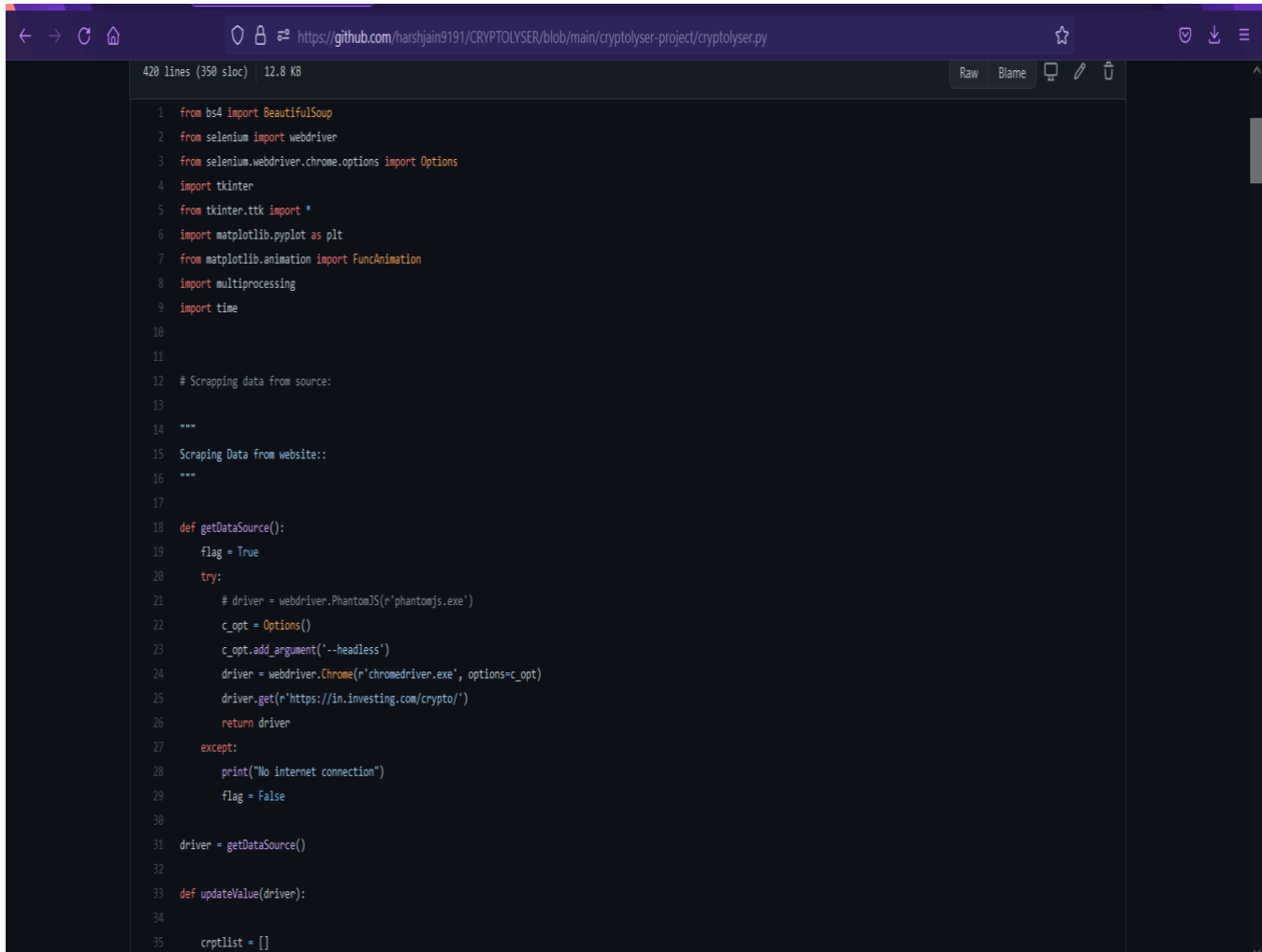
3. Real time plotting of data by Matplotlib:



4. The graph and GUI window running simultaneously in multiprocessing environment :



- Some snap out of the program codes is shown here as follows:

A screenshot of a web browser displaying a GitHub repository page for 'cryptolyser.py'. The browser's address bar shows the URL 'https://github.com/harshjain9191/CRYPTOLYSER/blob/main/cryptolyser-project/cryptolyser.py'. The file is 420 lines long, 350 sloc, and 12.8 KB. The code is written in Python and includes imports for BeautifulSoup, Selenium, Tkinter, and Matplotlib. It defines a 'getDataSource()' function that attempts to use PhantomJS or Chrome to scrape data from 'https://in.investing.com/crypto/'. The code also includes a comment about scraping data from a source and a list 'crptlist' at the bottom.

```
420 lines (350 sloc) | 12.8 KB
1  from bs4 import BeautifulSoup
2  from selenium import webdriver
3  from selenium.webdriver.chrome.options import Options
4  import tkinter
5  from tkinter.ttk import *
6  import matplotlib.pyplot as plt
7  from matplotlib.animation import FuncAnimation
8  import multiprocessing
9  import time
10
11
12  # Scrapping data from source:
13
14  ***
15  Scrapping Data from website::
16  ***
17
18  def getDataSource():
19      flag = True
20      try:
21          # driver = webdriver.PhantomJS(r'phantomjs.exe')
22          c_opt = Options()
23          c_opt.add_argument('--headless')
24          driver = webdriver.Chrome(r'chromedriver.exe', options=c_opt)
25          driver.get(r'https://in.investing.com/crypto/')
26          return driver
27      except:
28          print("No internet connection")
29          flag = False
30
31  driver = getDataSource()
32
33  def updateValue(driver):
34
35      crptlist = []
```

```
233 '''  
234 GUI Functions::  
235 '''  
236  
237 def show():  
238     global choice, bitcoin_value, value_now, buy_value  
239     if var.get() == 'bitcoin':  
240         bitcoin_value = updateValue(driver)[0]  
241         share_value['text'] = bitcoin_value  
242         choice = 1  
243         print(choice)  
244     elif var.get() == 'binance':  
245         bitcoin_value = updateValue(driver)[2]  
246         share_value['text'] = bitcoin_value  
247         choice = 3  
248         print(choice)  
249     elif var.get() == 'ethereum':  
250         bitcoin_value = updateValue(driver)[1]  
251         share_value['text'] = bitcoin_value  
252         choice = 2  
253         print(choice)  
254     elif var.get() == 'polkadot':  
255         bitcoin_value = updateValue(driver)[3]  
256         share_value['text'] = bitcoin_value  
257         choice = 4  
258         print(choice)  
259     elif var.get() == '1' or 1:  
260         share_value['text'] = "Please choose option"  
261  
262  
263 def buy():  
264     global var, buy_value, item  
265     item = var.get()  
266     info.config(text="Item Buy: "+str.upper(item), fg='green')  
267     var.set('1')  
268     buy_value = bitcoin_value
```



```
← → ↻ 🏠 https://github.com/harshjain9191/CRYPTOLYSER/blob/main/cryptolyser-project/cryptolyser.py ☆ 🔒 ⬇️ ☰
349 top_image = tkinter.PhotoImage(file='canvaltop.png')
350 top_image_label = Label(window, image=top_image, borderwidth=0).place(x=0, y=0)
351
352 # Heading label
353 label1 = tkinter.Label(window, text='Your Crypto-currency Analyser', fg='yellow', font=('Calibri Bold', 35),
354                        bg='#6fbf4c')
355 label1.place(x=80, y=45)
356
357 #Choice Label Heading
358 label2 = tkinter.Label(window, text='Choose your currency: ', font=('Sans Serif', 15)).place(x=60, y=180)
359
360 #Choices Radio button
361 rad_bitcoin = tkinter.Radiobutton(window, text='BITCOIN', variable=var, value='bitcoin',
362                                   font=('Arial Bold', 13), command=show).place(x=60, y=230)
363 rad_dogecoin = tkinter.Radiobutton(window, text='BINANCE', variable=var, value='binance',
364                                    font=('Arial Bold', 13), command=show).place(x=190, y=230)
365 rad_ethereum = tkinter.Radiobutton(window, text='ETHEREUM', variable=var, value='ethereum',
366                                    font=('Arial Bold', 13), command=show).place(x=330, y=230)
367 rad_polkadot = tkinter.Radiobutton(window, text='POLKADOT', variable=var, value='polkadot',
368                                    font=('Arial Bold', 13), command=show).place(x=460, y=230)
369
370 # number of share Label Heading and input
371 label3 = tkinter.Label(window, text='Enter number of shares: ', font=('Sans Serif', 15)).place(x=60, y=280)
372 input1 = tkinter.Spinbox(window, font=('Arial Bold', 15), from_=-1, to =10)
373 input1.place(x=300, y=280)
374
375 # value of per share
376 label4 = tkinter.Label(window, text='Share value: ', font=('Sans Serif', 15)).place(x=60, y=330)
377 share_value = tkinter.Label(window, text='00.00', font=('Sans Serif', 15))
378 share_value.place(x=180, y=330)
379
380 # BUY and SELL buttons
381 buy_butt = tkinter.Button(window, text='    BUY    ', font=('Arial Bold', 13), bg='forest green', borderwidth=0,
382                           fg='yellow', command=buy)
383 buy_butt.place(x=180, y=400)
384 sell_butt = tkinter.Button(window, text='    SELL    ', font=('Arial Bold', 13), bg='red3', borderwidth=0,
385                          fg='yellow', command=sell)
```

5. APPLICATIONS

✓ Advantages:

- I. Cryptolyser is a very light-weight software, and requires minimal resources to run on any device which supports python shell.
- II. It is easy to use and very friendly GUI.
- III. It has features to look around with a real time simulation.
- IV. The real time measuring graph help user to understand at ease.

✓ Disadvantages:

- I. Being a standalone software, it requires a fixed version of python and chrome browser to work on.
- II. It is based on simulation, not to be treated as real world entity.
- III. It lacks certain accounting features.

✓ Applications:

- I. It is a simulation tool for checking the crypto markets.
- II. It is an analysis tool which calculates both losses and profits of share values, when a crypto is bought.
- III. It consist of graphs which helps user to visualize the crypto currencies up – downs with real time.

6.CONCLUSION AND FUTURE WORK:

Cryptolyser is a light – analysis tool for analysing Crypto – currencies market. It is a easy to use software which runs on python shell and scripting procedures. It calculates share prices, deals profits or losses transactions and provides graphs to visualize data.

Cryptolyser is brought to user as an alternative to heavy software to see through crypto market and reach to user with an efficient way to calculate the profits with there respective currencies, in a simulated environment where they can understand the nature and frequently changing share values of crypto – currencies and to monitor them in almost and accurate real time.

Cryptolyser, as a simple and light weighted, has a impactful future to be as an upgraded software and as a packages. Proper accounting section with database connectivity could be introduced in later versions with APIs to introduce it with real world crypto – money transactions and a better visualization to buy or sale crypto – currencies online without any third party application.

7.REFERENCES:

1. Google Searches www.google.com/in
2. Python docs <https://docs.python.org/3/>
3. Github <https://github.com/harshjain9191/CRYPTOLYSER>
4. Matplotlib docs <https://matplotlib.org/>
5. Online graphic designs <https://www.befunky.com/features/graphic-designer/>