BINUS INTERNATIONAL

ALGORITHM AND PROGRAMMING

Final Project Report

CryptOne

Submitted By:

Jovan Nikholas (2902641811)

Class L1AC

Computer Science Program

Table of Contents

1. Background

Cryptocurrency is a digital currency that is in a trend right now. Unlike, usual currency there is no physical money that we could use offline(Montevirgen, n.d.). All of Cryptocurrency is in a form of digital money, that could be accessed by anyone. By using Cryptocurrency we could do the transaction more secure and our privacy is safer than normal currency.

The difference between Cryptocurrency and normal currency is it is not been controlled by the government(Gowda & Chakravorty, 2021). Because it is not controlled by government, people have full control of they money. They could do transactions any time without others knowing. For a normal currency, once a transactions are done the government or bank keep record of the data and could use the data for they own will. By not controlled by the government it does not have anything related by political influences(Amon, 2025). For example, the price of a normal currency like USD or IDR are very dependent by the political influences. The value could go up and downs based on the news that the world is facing right now.

Similar to the normal currency, crypto has a lot of variation. If a normal currency every single country has a different kind, crypto also. They also have a lot of variation from the stable coin to alt coin(Dewani et al., 2020). Stable coin such as Bitcoin or Ethereum are stable because they tend to hold their value. Usually, All of crypto price are the direct result of the bitcoin price, just like the normal currency price is depends on the price of USD. However, for alt coins sometimes their price does not depend on anything. Alt coin is a coin who is sometimes made by a small group of people.

Just like anything else there is advantages and disadvantages of something. In crypto for the advantages are:

1. Fast Transactions. It is fast transactions, because the scale of crypto is global so anyone could do it anytime and anywhere. A normal currency transaction depends on the work hour (bank). If it is not in working hours, we could not do transactions.
2. The fee of transactions is low compared to bank(MURRY CIERRA, 2025). By using crypto, our transactions fee is lower because it is global after all. All you need to pay is only the network fees. Compare this to normal bank transactions, if you wanted to do transaction overseas the bank charge a lot of money just to do transactions.

Those are the advantages of Cryptocurrency, the disadvantage are :

1. Prices are volatile(Delfabbro et al., 2021). The means of “Volatile” is the price could go up and downs very fast. Imagine if you are a trader, for a normal stocks there is like stock hours and the price difference at max per day is around 50%. For crypto, there is no stock hours it always up and running for 24/7. Then the crypto price could go up and down from 100 to 1000%. This is

especially in alt coins. An alt coin value could jump rocket up or down, so it is very volatile.

2. Cryptocurrency could be misuse for illegal activities. Because it is not controlled by government, sometimes crypto transactions are use to sell illegal stuffs. It is because the government could not see the transactions at all.

Similar to stocks, to do a transaction a financial app is needed. The app itself called securities. In this app there is a lot of functionality such as viewing Crypto Currency (price, daily change, history) and even do transaction (deposit, withdrawal, transfer). The problem for this app however, they sometimes required us to do sign up. Usually, the data for the signup is our name, email, password, birthdate and for Indonesian citizen they require us to submit our Identity Card (KTP). Sometimes an app could sell this data to the third party. By submitting our Identity Card, if the app is not trust worthy they could misuse our data such as scamming, money fraud.

2. Problem description

As stated in background, by submitting our data to shady securities they could misuse our data. Scamming, money fraud are types of fraud that often happens. How they do it by selling our data to shady institution. From there, they could sometimes contact us pretending to be some institutions (bank, government). By pretending they could trick us into sending money to them.

Sometimes, people only wanted to see crypto updates like prices, history, conversion crypto and crypto and conversion to real currency. If the things that needed to do are this, it is a hassle to register your data. For the solution of this problem, I am trying to develop an app called CryptOne. This app is all in one app to view crypto related things.

I've designed this program so it could be use by anyone, and trying to simplify to do things. I also made this program for my own use also, because I am also a crypto enthusiast after all.

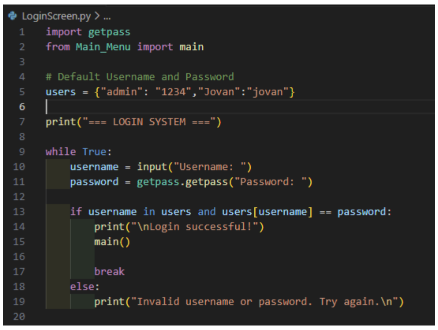
This program is using the API from the internet called "freecryptoapi". I am patching the data from there and translate it and format it into more understandable format. From the API the data is in the format of JSON, and the layout is a little bit hard to see.

Because it is utilizing an API, a license key is needed. For this website "freecryptoapi" the API itself is free, but there is some function that we require to pay. Because, personally I am going to be use this app, I pay the small fee to unlock the special features

3. Project Specification

For the project specification below are the code and the explanation for each python file that has been created.

3.1 LoginScreen.py

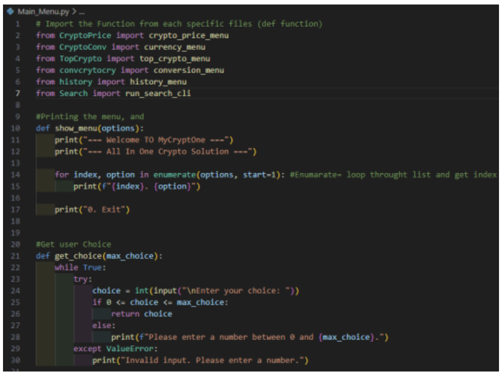


```
LoginScreen.py > ...
import getpass
from Main_Menu import main

# Default Username and Password
users = {"admin": "1234","Jovan":"jovan"}

print("=== LOGIN SYSTEM ===")

while True:
    username = input("Username: ")
    password = getpass.getpass("Password: ")

    if username in users and users[username] == password:
        print("\nLogin successful!")
        main()

        break
    else:
        print("Invalid username or password. Try again.\n")
```

This is the code for my login screen. In here I am using the imported module from python "getpass". By using "getpass" when we type our password the character will not be visible, enhancing the security of the program. Then for the username and password I store this in dictionary data structure, it is because dictionary is a keypairs so it stores matching pair like {username:password}.

Below that is a simple logic of if statement that states if the username and password matches the credential it will run the main(). Main() is a python function that executes the main dashboard in the system, however if the username and password did not match the main dashboard will not open.

3.2 Main_Menu.py



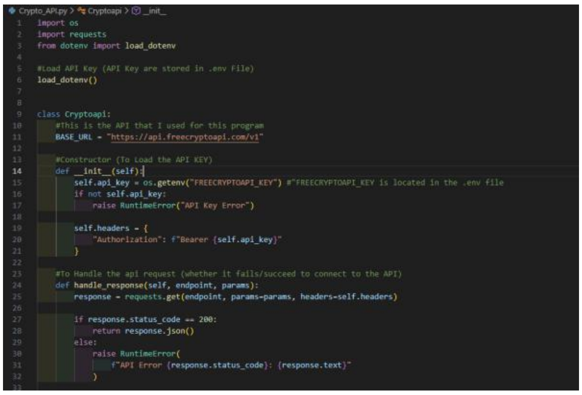
```
Main_Menu.py > ...
1   # Import the Function from each specific files (def function)
2   from CryptoPrice import crypto_price_menu
3   from CryptoConv import currency_menu
4   from TopCrypto import top_crypto_menu
5   from convcrytocry import conversion_menu
6   from history import history_menu
7   from Search import run_search_cli
8
9   #Printing the menu, and
10  def show_menu(options):
11      print("--- Welcome TO MyCryptOne ---")
12      print("--- All In One Crypto Solution ---")
13
14      for index, option in enumerate(options, start=1): #Enumarate- loop throught list and get index
15          print(f"{index}. {option}")
16
17      print("0. Exit")
18
19
20  #Get user Choice
21  def get_choice(max_choice):
22      while True:
23          try:
24              choice = int(input("\nEnter your choice: "))
25              if 0 <= choice <= max_choice:
26                  return choice
27              else:
28                  print(f"Please enter a number between 0 and {max_choice}.")
29          except ValueError:
30              print("Invalid input. Please enter a number.")
```

```
#Main Menu
def main():
    options = [
        "Crypto Price",
        "Crypto Converter (EX : BTC to IDR)",
        "Top Crypto List","Crypto Converter(EX : BTC to ETH)","Crypto History","Search"
    ]

    while True:
        show_menu(options)
        choice = get_choice(len(options))

        if choice == 0:
            print("Exiting program...")
            break

        print(f"You selected: {options[choice - 1]}")

        if choice == 1:
            crypto_price_menu()
        if choice ==2:
            currency_menu()
        if choice ==3:
            top_crypto_menu()
        if choice ==4:
            conversion_menu()
        if choice ==5:
            history_menu()
        if choice ==6:
            run_search_cli()

if __name__ == "__main__":
    main()
```

This are the code for my Main Menu. We are going to start with images on the left. Line 1-7 are all imported def function from other python files so when a menu is executed it will run the def function from another python file. For the line 10-17 it is a def function to print numbers in the menu automatically. It uses an enumerate as a python built in function for automatic number numbering. Moving on to line 21-30 it is the def function to validate user choice, if the user choice is correct according to the menu if it is correct it will open the app function that they are going to choose.

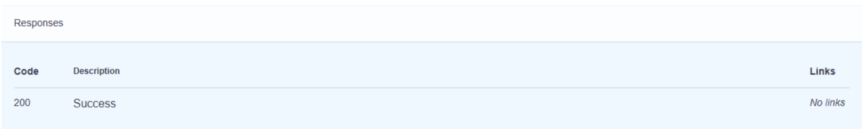
For the right image the def main() it is the main function that display and handle all of the user input an printing the menu. I am storing all of the menu in a list, so later I could change the menu if I wanted to. From there, “While” function is used to display the menu continuously for the user to input the function, to break the while loop, pressing 0 will terminate it because it is closing the program. When the user inputted the correct numbers, it will run the def function that is in the another python file.

3.3 Crypto_API.py and CryptoList.py



```
Crypto_API.py > Cryptoapi > __init__
1  import os
2  import requests
3  from dotenv import load_dotenv
4
5  #Load API Key (API Key are stored in .env File)
6  load_dotenv()
7
8
9  class Cryptoapi:
10     #This is the API that I used for this program
11     BASE_URL = "https://api.freecryptoapi.com/v1"
12
13     #Constructor (To Load the API KEY)
14     def __init__(self):
15         self.api_key = os.getenv("FREECRYPTOAPI_KEY") #"FREECRYPTOAPI_KEY is located in the .env file
16         if not self.api_key:
17             raise RuntimeError("API Key Error")
18
19         self.headers = {
20             "Authorization": f"Bearer {self.api_key}"
21         }
22
23     #To Handle the api request (whether it fails/succeed to connect to the API)
24     def handle_response(self, endpoint, params):
25         response = requests.get(endpoint, params=params, headers=self.headers)
26
27         if response.status_code == 200:
28             return response.json()
29         else:
30             raise RuntimeError(
31                 f"API Error {response.status_code}: {response.text}"
32             )
33
```

The purposes of this python file is to handle all of the API request that will be made in this program. I am using python imported module such as “os”, “requests” and “dotenv”. The function of os is to make python find a directory. It is used to find the “dotenv” file that stores this program’s API key. The API key it self is stored in another file, so not everyone could use this program. They have to have the API KEY. The “requests” are used to send request to the API websites. For line 14-20 it functions to find my API key using “os” then match it with the API Sites. For the line 24-32 are ways to validate if the params that we send to the API is correct or isn’t, besides that is also to check if our API receive our request or not. If it receive our request it will give the code 200. The code 200 is from the API itself (below is the attached image from the API site.)



Responses

Code	Description	Links
200	Success	*No links*

```python
class Cryptoapi:
    #Getter Function to get Crypto Data from the API (Used for the 1st menu in the menu page)
    def get_crypto_data(self, symbol):
        endpoint = f"{self.BASE_URL}/getData" #API Link
        params = {"symbol": symbol} #parameter that are required to fetch the data
        return self.handle_response(endpoint, params)

    #Getter Function to convert CryptoCurrency to Real Currency
    def get_data_currency(self, symbol: str, local: str):
        endpoint = f"{self.BASE_URL}/getDataCurrency"
        params = { "symbol": symbol,"local": local}
        return self.handle_response(endpoint, params)

    #Getter Function to See The Top Crypto
    def get_top_crypto(self, top):
        endpoint = f"{self.BASE_URL}/getTop"
        params = {"top": top}
        return self.handle_response(endpoint, params)

    #Getter Function to convert From 1 crypto to another Crypto
    def get_conversion(self, from_symbol: str, to_symbol: str, amount: int):
        endpoint = f"{self.BASE_URL}/getConversion"
        params = {"from": from_symbol,"to": to_symbol,"amount": amount}
        return self.handle_response(endpoint, params)

    #Getter Function to get history from selected crypto in whatever periods
    def get_history(self,symbol: str,days: int):
            endpoint = f"{self.BASE_URL}/getHistory"
            params = {"symbol": symbol,"days":days}
            return self.handle_response(endpoint, params)
```

The image above show all of the function that the API supported that I used to build my application. It is a getter method that returns all the data that we intended a request. To make use of this getter method for each function we need to specify several things. I'm going to explain for the get_crypto_data and for the rest is similar. First, we need to specify our API link. For the get_crypto_data the link is /getData. Actually the base.url has been set in the first image which is "https://api.freecryptoapi.com/v1". So, combined link is "https://api.freecryptoapi.com/v1/getData". Then we need to specify the params that the we have to fulfil for the API to work, in this case is symbol. After that we connect to the API, using def handle_response(image 1). Below attached is the documentation from the API website for /getData which contains the params. The rest for the getter function is similar, the only different is the params and the API links.

GET **/getData** Get single or multiple crypto currency data

Parameters | Try it out

Name	Description
symbol * required string *(query)*	Symbols separated by + (e.g., BTC+ETH+ETHBTC@binance) symbol

Responses

Code	Description	Links
200	Success	*No links*

```python
class Search:

    #Constructor for searching crypto in Dictionary
    def __init__(self, crypto_dict, currency_dict):
        self.crypto_dict = crypto_dict
        self.currency_dict = currency_dict

    #Function for searching crypto
    def search_symbol(self, dictionary, user_input):
        user_input = user_input.lower()

        #Based on the intital (EX : BTC)
        for symbol, fullname in dictionary.items():
            if user_input == symbol.lower():
                return f"{symbol} → {fullname}"

        #Based on full name
        for symbol, fullname in dictionary.items():
            if user_input == fullname.lower():
                return f"{fullname} → {symbol}"

        #Based on partial fullname
        for symbol, fullname in dictionary.items():
            if user_input in fullname.lower():
                return f"{fullname} → {symbol}"

        #Based on partial symbol
        for symbol in dictionary:
            if user_input in symbol.lower():
                return f"{symbol} → {dictionary[symbol]}"

        return "No matching result found."

    # Search by crypto
    def find_crypto(self, query):
        return self.search_symbol(self.crypto_dict, query)

    # Search by currency
    def find_currency(self, query):
        return self.search_symbol(self.currency_dict, query)
```

CryptoList.py > CRYPTO_TOKENS

```python
CRYPTO_TOKENS = {
    "BTC": "Bitcoin",
    "ETH": "Ethereum",
    "USDT": "Tether",
    "USDC": "USD Coin",
    "BNB": "Binance Coin",
    "XRP": "XRP",
    "ADA": "Cardano",
    "DOGE": "Dogecoin",
    "SOL": "Solana",
    "DOT": "Polkadot",
    "TRX": "TRON",
    "MATIC": "Polygon",
    "LTC": "Litecoin",
    "BCH": "Bitcoin Cash",
```

The two image above (left: search class ,right : Crypto/currency list).

The image left is a function to search crypto/currency abbreviation that is stored in the list (right). The search is capable of searching by initial or full name.

3.4 CryptoPrice.py

```python
from Crypto_API import Cryptoapi

#Def Function to used a choice if wanted to go back to the main menu
def return_to_menu(menu_name="Main Menu"):
    choice = input(f"\nReturn to {menu_name}? (Y/N): ").strip().lower()
    return choice == "y"

def format_crypto_data(data):
    if data.get("status") != "success":
        return "Failed to get data."

    symbol_info = data["symbols"][0]

    formatted = (
        f"\n=== Crypto Information ===\n"
        f"Symbol        : {symbol_info['symbol']}\n"
        f"Last Price    : {symbol_info['last']} USD\n"
        f"Lowest Price  : {symbol_info['lowest']} USD\n"
        f"Highest Price : {symbol_info['highest']} USD\n"
        f"24h Change    : {float(symbol_info['daily_change_percentage']):.2f}%\n"
        f"Source        : {symbol_info['source_exchange']}\n"
        f"Last Updated  : {symbol_info['date']}\n"
        f"=========================\n"
    )
    return formatted
```

```python
#Function to Run the crypto Price
def crypto_price_menu():
    api = Cryptoapi()

    while True:
        symbol = input("Enter crypto symbol (ex: BTC, ETH, DOGE): ").upper()

        try:
            data = api.get_crypto_data(symbol)
            print(format_crypto_data(data))
        except Exception as e:
            print("Error:", e)

        if return_to_menu("Main Menu"):
            break
```

This is the Screenshot of my code for the function of finding crypto prices. Starting by the image on the right. The image on the right is the function to find the crypto data by using the user input. From there the data is patched to the Crypto_API.py, def get_crypto_data. Then it will return the Json Format such as {'status': 'success', 'symbols': [{'symbol': 'BTC', 'last': 1531958322.5128, 'last_btc': '1', 'lowest': 1486988257.5452, 'highest': 1540544447.0159998, 'date': '2025-12-08 16:01:20', 'daily_change_percentage': '0.36179294970727'}]}. Because the data presentation is not good the data got formatted into something more usable in def format_crypto (Left Image). Also when the users has finished looking for the information, it will ask the user to return to the main menu by typing y/n (y to go back to main menu, n to do the function once again)

3.5 CryptoConv.py

```python
CryptoConv.py > format_currency_data
from Crypto_API import Cryptoapi

def return_to_menu(menu_name="Main Menu"):
    choice = input(f"\nReturn to {menu_name}? (Y/N): ").strip().lower()
    return choice == "y"


def format_currency_data(data, local="USD"):
    if data.get("status") != "success":
        return "Failed to get currency data."

    coin = data["symbols"][0]  # API always returns a list

    # If there is None Values
    last = coin["last"]
    lowest = coin["lowest"] if coin["lowest"] is not None else "N/A"
    highest = coin["highest"] if coin["highest"] is not None else "N/A"

    # Some tokens have no 24h change (USDT, USDC)
    try:
        change = float(coin["daily_change_percentage"])
        change = f"{change:.2f}%"
    except:
        change = "N/A"

    return (
        "\n----- Crypto Currency Data -----\n"
        f"Symbol        : {coin['symbol']}\n"
        f"Last Price    : {last} {local}\n"
        f"Lowest Price  : {lowest} {local}\n"
        f"Highest Price : {highest} {local}\n"
        f"24h Change    : {change}\n"
        f"Last Updated  : {coin['date']}\n"
        "================================\n"
    )
```

```python
def currency_menu():
    api = Cryptoapi()

    while True:
        symbol = input("Enter crypto symbol (ex: BTC, ETH): ").upper()
        local = input("Enter Currency:  ").upper()
        try:
            data = api.get_data_currency(symbol, local)
            print(format_currency_data(data, local))
        except Exception as e:
            print("Error:", e)
        if return_to_menu("Main Menu"):
            break
```

This is the Screenshot of my code for the function for converting crypto prices into currency. Starting by the image on the right. The image on the right is the function to convert crypto price into currency. From there the data is patched to the Crypto_API.py, get_data_currency. Then it will return the Json Format such as {'status': 'success', 'symbols': [{'symbol': 'BTC', 'last': 1502682704.6399999, 'last_btc': '1', 'lowest': 1491310566.72, 'highest': 1546783145.6000001, 'date': '2025-12-11 15:54:43', 'daily_change_percentage': '-2.4796083522579'}]}.Because the data presentation is not good the data got formatted into something more usable in def format_currency_data (Left Image). Also,when the users has finished looking for the information, it will ask the user to return to the main menu by typing y/n (y to go back to main menu, n to do the function once again)

3.6 TopCrypto.py

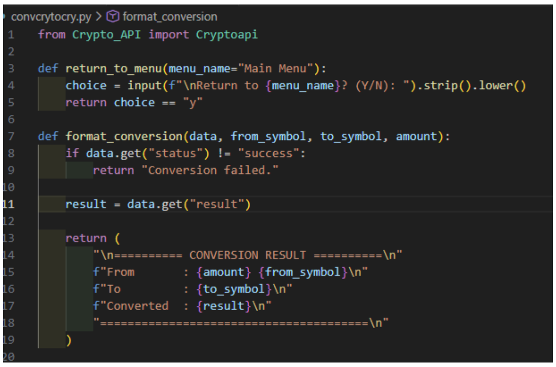
```python
from Crypto_API import Cryptoapi

def return_to_menu(menu_name="Main Menu"):
    choice = input(f"\nReturn to {menu_name}? (Y/N): ").strip().lower()
    return choice == "y"

def format_top_crypto(data):
    if not data.get("status"):
        return  "Fail to get data."

    output = "\n========== TOP CRYPTOCURRENCIES ==========\n"

    #Loop throught the json file
    for coin in data["symbols"]:
        symbol = coin["symbol"]
        rank = coin.get("rank", "-")

        price = coin["last"]
        change = coin["daily_change_percentage"]

        lowest = coin["lowest"] if coin["lowest"] is not None else "N/A"
        highest = coin["highest"] if coin["highest"] is not None else "N/A
        change_str = f"{change:.2f}%" if change is not None else "N/A"

        output += (
            f"{rank:>2}. {symbol:<5} | "
            f"Price: {price:<12} | "
            f"24h Change: {change_str:<8} | "
            f"Low: {lowest:<10} | "
            f"High: {highest:<10}\n"
        )

    output += "==========================================\n"
    return output
```

```python
def top_crypto_menu():
    api = Cryptoapi()

    while True:
        # Ask for top count
        while True:
            try:
                top = int(input("How many top cryptos to display? (ex: 5, 10, 20): "))
                if top > 0:
                    break
                print("Enter a number greater than 0")
            except ValueError:
                print("Please enter a valid number.")

        print("\nLoading Top Cryptocurrencies...\n")

        try:
            data = api.get_top_crypto(top)
            print(format_top_crypto(data))
        except Exception as e:
            print("Error:", e)

        if return_to_menu("Main Menu"):
            break
```

This is the Screenshot of my code for the function for seeing the trending crypto currency.Starting by the image on the right. The image on the right is the function to seeing the trending crypto currency. It ask the users to input how many crypto that the user wishes to display by using the user input. To validate while loop is used to ensure the users only input integer format. Then when the user has entered the data it will patched the data to Crypto_API.py, def get_top_crypto. Then it will return to the Json format such as {'status': True, 'symbols': [{'symbol': 'BTC', 'rank': 1, 'last': 90100.21, 'last_btc': 1, 'lowest': 89450.01, 'highest': 92777.3, 'date': '2025-12-11 16:03:11', 'daily_change_percentage': -2.51413406515, 'source_exchange': 'binance'}, {'symbol': 'ETH', 'rank': 2, 'last': 3203.41, 'last_btc': 0.035553857199667, 'lowest': 3171.18, 'highest': 3365.6, 'date': '2025-12-11 16:03:11', 'daily_change_percentage': -4.4622326671697, 'source_exchange': 'binance'}, {'symbol': 'USDT', 'rank': 3, 'last': 1.0001014818449, 'last_btc': None, 'lowest': None, 'highest': None, 'date': '2025-12-11 16:03:12', 'daily_change_percentage': None, 'source_exchange': None}], 'count': 3, 'loaded_time': 0.31425}.

To make the data more readable, the data is processed in def format_top_crypto. Because users could specify of how many crypto does the users need, the formatter needed to be dynamic. So it uses for loop to get a loop of every data, and making the list. Because, when testing some of the data did not have lowest , highest and change % , we have to tell the formatter to make it as n/a in the final product.

3.7 Convcrytocry.py



convcrytocry.py > format_conversion

```python
from Crypto_API import Cryptoapi

def return_to_menu(menu_name="Main Menu"):
    choice = input(f"\nReturn to {menu_name}? (Y/N): ").strip().lower()
    return choice == "y"

def format_conversion(data, from_symbol, to_symbol, amount):
    if data.get("status") != "success":
        return "Conversion failed."

    result = data.get("result")

    return (
        "\n========== CONVERSION RESULT ==========\n"
        f"From       : {amount} {from_symbol}\n"
        f"To         : {to_symbol}\n"
        f"Converted  : {result}\n"
        "=======================================\n"
    )
```

```python
def conversion_menu():
    api = Cryptoapi()

    while True:

        from_symbol = input("Input the crypto that you want to convert (ex: BTC, ETH, Doge): ").upper()
        to_symbol = input("Input the crypto you want to convert  (ex: BTC, ETH, Doge): ").upper()

        try:
            amount = float(input("Amount to convert: "))
        except ValueError:
            print("Enter a valid number for amount.")
            continue

        print("\nConverting...\n")

        try:
            data = api.get_conversion(from_symbol, to_symbol, amount)
            print(format_conversion(data, from_symbol, to_symbol, amount))
        except Exception as e:
            print("Error:", e)

        if return_to_menu("Main Menu"):
            break
```

This is the Screenshot of my code for the function for converting crypto to crypto with a specific amount. The image on the right is the function to convert it. It ask the user for the input from what crypto, to what crypto and the amount that they wish to convert. It uses a while loop to make sure that the data that the user input is in the correct format. From there the data will be patched to Crypto_API.py, def get_conversion. It will return the data as Json Format such as. {'status': 'success', 'result': 28.14347687480075}. To make the data readable it will format the data in def format_conversion.

3.8 History.py

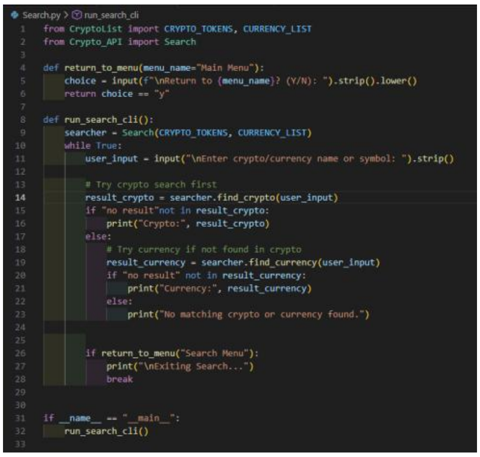
```python
from Crypto_API import Cryptoapi

def return_to_menu(menu_name="Main Menu"):
    choice = input(f"\nReturn to {menu_name}? (Y/N): ").strip().lower()
    return choice == "y"

def format_history(data):
    if data.get("status") != "success":
        return "Failed to get history data."

    entries = data.get("symbols", [])

    if not entries:
        return "No history found."

    output = "\n========= CRYPTO PRICE HISTORY =========\n"
    output += f"{'Date':<22} {'Close Price (USD)':<15}\n"
    output += "-" * 40 + "\n"

    #'Date':<22 max character 22 left aligned
    #{'Close Price (USD)':<15}\n"  second column max 15 character

    for entry in entries:
        output += f"{entry['date']:<22} {entry['close']}\n"

    output += "========================================\n"
    return output

```

```python
def history_menu():
    api = Cryptoapi()

    while True:
        symbol = input("\nEnter crypto symbol for history (ex: BTC, ETH): ").upper()

        try:
            limit = int(input("Enter history days(ex: 5, 10, 30): "))
        except ValueError:
            print("Please enter a valid number.")
            continue

        print("\nFetching history...\n")

        try:
            data = api.get_history(symbol, limit)
            print(format_history(data))
        except Exception as e:
            print("Error:", e)

        if return_to_menu("Main Menu"):
            break

```

This is the Screenshot of my code to display specific crypto history within the specified days. The image on the right is the function to do see the crypto history. It have the user input to input what crypto they want to convert and how many days of history the user wants. It uses a while loop to ensure that the user given the data is valid. From there the data is patched into Crypto_API.py,def get_history. Then it will returns the data as json format such as #{'status': True, 'symbols': [{'symbol': 'BTC', 'rank': 1, 'last': 90100.21, 'last_btc': 1, 'lowest': 89450.01, 'highest': 92777.3, 'date': '2025-12-11 16:03:11', 'daily_change_percentage': -2.51413406515, 'source_exchange': 'binance'}, {'symbol': 'ETH', 'rank': 2, 'last': 3203.41, 'last_btc': 0.035553857199667, 'lowest': 3171.18, 'highest': 3365.6, 'date': '2025-12-11 16:03:11', 'daily_change_percentage': -4.4622326671697, 'source_exchange': 'binance'}, {'symbol': 'USDT', 'rank': 3, 'last': 1.0001014818449, 'last_btc': None, 'lowest': None, 'highest': None, 'date': '2025-12-11 16:03:12', 'daily_change_percentage': None, 'source_exchange': None}], 'count': 3, 'loaded_time': 0.31425}.

Because users could specify of how many days does the users need to view the history, the formatter needed to be dynamic. So, it uses for loop to get a loop of every data, and making the list. Because, when testing some of the data did not have lowest , highest and change % , we have to tell the formatter to make it as n/a in the final product.

3.9 Search.py



Search.py > run_search_cli

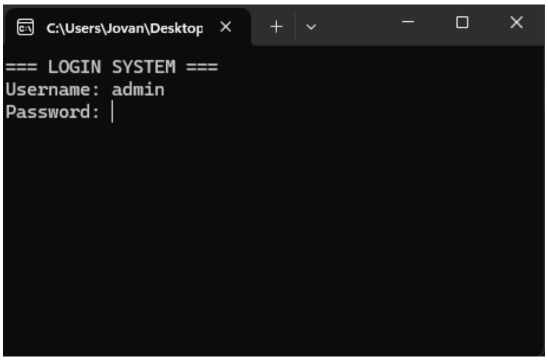
```python
from CryptoList import CRYPTO_TOKENS, CURRENCY_LIST
from Crypto_API import Search

def return_to_menu(menu_name="Main Menu"):
    choice = input(f"\nReturn to {menu_name}? (Y/N): ").strip().lower()
    return choice == "y"

def run_search_cli():
    searcher = Search(CRYPTO_TOKENS, CURRENCY_LIST)
    while True:
        user_input = input("\nEnter crypto/currency name or symbol: ").strip()

        # Try crypto search first
        result_crypto = searcher.find_crypto(user_input)
        if "no result"not in result_crypto:
            print("Crypto:", result_crypto)
        else:
            # Try currency if not found in crypto
            result_currency = searcher.find_currency(user_input)
            if "no result" not in result_currency:
                print("Currency:", result_currency)
            else:
                print("No matching crypto or currency found.")


        if return_to_menu("Search Menu"):
            print("\nExiting Search...")
            break


if __name__ == "__main__":
    run_search_cli()
```

This is the screenshot of search function in the program. First the search database (CRYPTO_TOKENS,CURRENCY_LIST) is imported into python dan from the Crypto_API.py, Search class is imported. The ways the search works is user inputted what they want to search. The search function first will find in the crypto_tokens database if the users is searching for crypto or not, if it could not find any it will dump it to currency_list.

4. Program Features :

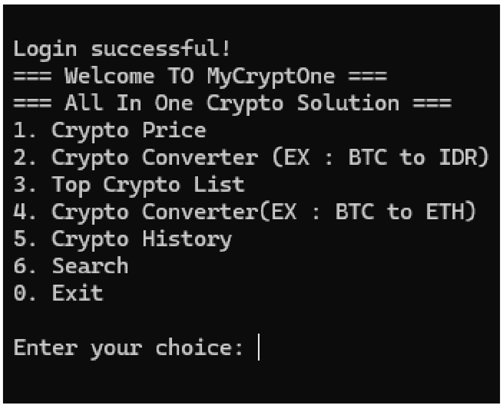
Below are the program functionality and screen shot of the CryptOne Program.

4.1 Login Screen



For added security, our program includes login screen that has attributes of username and password. If a user type different credential that is not in the system, it will not log the users in and will not open the main dashboard

4.2 Main Dashboard



Main menu serves as the default interface when the user correctly type the credential. In this menu there are 6 functions that the users could choose : Crypto Price, Crypto Converter(BTC to IDR), Top Crypto List, Crypto Converter(BTC to ETH), Crypto History, Search and Exit

4.3 Crypto Price

```
Enter your choice: 1
You selected: Crypto Price
Enter crypto symbol (ex: BTC, ETH, DOGE): BTC

=== Crypto Information ===
Symbol        : BTC
Last Price    : 90070.11 USD
Lowest Price  : 89450.01 USD
Highest Price : 92777.3 USD
24h Change    : -2.55%
Source        : binance
Last Updated  : 2025-12-11 16:56:13
==========================


Return to Main Menu? (Y/N): |
```

The function of a crypto price is to display user desired crypto information. It will display information such as Price(Lowest,Highest,Last Price) , 24 H Price, Source (Where is the data from), Last Updated

4.4 Crypto Converter (EX : BTC to IDR)

```
Enter your choice: 2
You selected: Crypto Converter (EX : BTC to IDR)
Enter crypto symbol (ex: BTC, ETH): btc
Enter Currency:  idr

===== Crypto Currency Data =====
Symbol        : BTC
Last Price    : 1501178293.0847998 IDR
Lowest Price  : 1491333823.7225997 IDR
Highest Price : 1546807267.698 IDR
24h Change    : -2.58%
Last Updated  : 2025-12-11 17:11:02
================================


Return to Main Menu? (Y/N): |
```

The function of crypto converter is to convert crypto prices into real currency. To use this function users need to input they desired crypto and the currency they want to convert. After converting the data that will be displayed are price (lowest price,Highest Price, Last Price), 24 h changed and last updated.

4.5 Top Crypto List

```
Enter your choice: 3
You selected: Top Crypto List
How many top cryptos to display? (ex: 5, 10, 20): 5

Loading Top Cryptocurrencies...


========== TOP CRYPTOCURRENCIES ==========
 1. BTC   | Price: 90160          | 24h Change: -2.45%  | Low: 89450.01   | High: 92777.3
 2. ETH   | Price: 3193           | 24h Change: -4.77%  | Low: 3171.18    | High: 3365.6
 3. USDT  | Price: 1.0003006616867 | 24h Change: N/A      | Low: N/A        | High: N/A
 4. XRP   | Price: 2.0099         | 24h Change: -2.58%  | Low: 1.9943     | High: 2.0687
 5. BNB   | Price: 868.15         | 24h Change: -3.40%  | Low: 862.09     | High: 901.51
==========================================


Return to Main Menu? (Y/N):
```

The function of top crypto list is to see what is the trending crypto currency right now. To use this function user need to specify how many trending crypto they want to see. After inputting of how many crypto they wanted to see, the result that is displayed are (List Trending Crypto, price(High,Low,Last) and 24h change.

4.6 Crypto Converter (EX : BTC to ETH)

```
Enter your choice: 4
You selected: Crypto Converter(EX : BTC to ETH)
Input the crypto that you want to convert (ex: BTC, ETH, Doge): BTC
Input the crypto you want to convert  (ex: BTC, ETH, Doge): ETH
Amount to convert: 1

Converting...


========= CONVERSION RESULT =========
From       : 1.0 BTC
To         : ETH
Converted  : 28.229504913312887
=====================================


Return to Main Menu? (Y/N):
```

The function of this crypto converter is to convert from One crypto to another with specified amount. To use this function user need to specific of what crypto they want to convert and to the crypto that they want to convert and the amount. After inputting of it , the result of that conversion will be displayed.

4.7 Crypto History

```
Enter your choice: 5
You selected: Crypto History

Enter crypto symbol for history (ex: BTC, ETH): BTC
Enter history days(ex: 5, 10, 30): 5

Fetching history...


========= CRYPTO PRICE HISTORY =========
Date                     Close Price (USD)
----------------------------------------
2025-12-10 23:59:59    92423.87
2025-12-09 23:59:59    93112.76
2025-12-08 23:59:59    90795.83
2025-12-07 23:59:59    91439.04
2025-12-06 23:59:59    89548.88
========================================


Return to Main Menu? (Y/N):
```

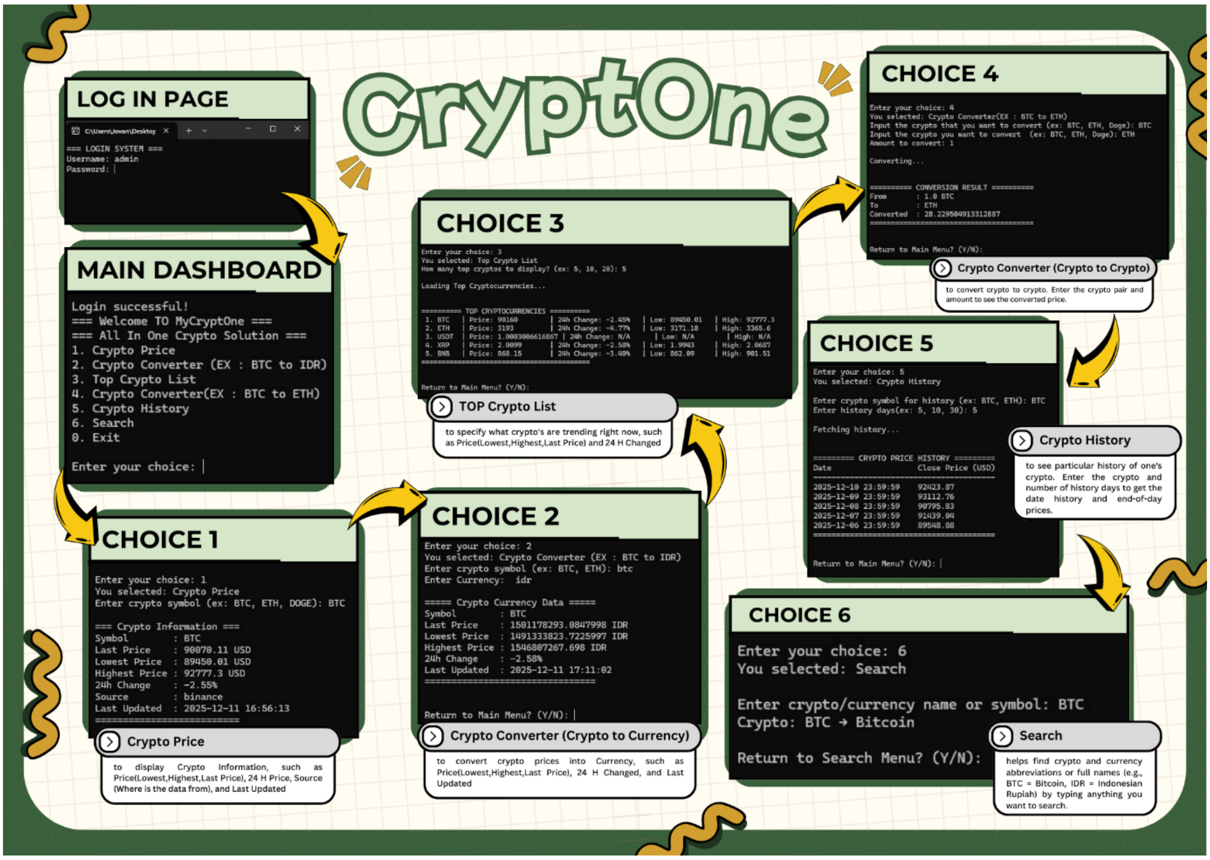
The function of crypto history is to find desired crypto history for the specified days. To use this function, users need to input their desired crypto and how many days the users want to see the history. After inputting it, it will display the last crypto price at that day.

4.8 Search

```
Enter your choice: 6
You selected: Search

Enter crypto/currency name or symbol: BTC
Crypto: BTC → Bitcoin

Return to Search Menu? (Y/N):
```

The function of the search is to find cryptocurrency or real currency for their real name or the abbreviation.

5. Project Posters



6. Appendix

1. Files related to Project (Project Code,Project Report):

https://github.com/jovannik88/Algorithm-and-Programming-Sem-1-Final-Project-CryptOne-

2. Step by Step Running the Program :

    1. Download a folder called "Cryptone exe folder"

    2. Make sure to put the exe and env file into single folder

    3. Run it

    4. Enter Credential

    5. Feel Free to explore Cryptone Function

    6. Default Username and Password : “Admin” , “1234”

3. AI Declaration Statement :

To help me on making this project, I am using AI to help me sharing the ideas for this project. The AI that I use are :

1. ChatGPT : I am using Chat GPT to review my code, and giving the suggestion to my code, and learning how to import the API into Phyton

2. Gemini : I am using Gemini to help me review my code.

7. References

Amon. (2025, November 25). *Cryptocurrency vs Fiat Currency: The Fundamental Differences*. Cisin. https://www.cisin.com/coffee-break/the-fundamental-differences-between-cryptocurrency-and-fiat-currency.html?utm_source=chatgpt.com

Delfabbro, P., King, D., Williams, J., & Georgiou, N. (2021). Cryptocurrency trading, gambling and problem gambling. *Addictive Behaviors*, *122*. https://doi.org/10.1016/j.addbeh.2021.107021

Dewani, J., Ghodawat, M., Vadhera, S., Patel, M., Mishra, A., & Kotha, N. (2020). A research study on awareness regarding crypto currency among investors. *International Journal on Integrated Education*, *3*(III), 114–125.

Gowda, N., & Chakravorty, C. (2021). Comparative study on cryptocurrency transaction and banking transaction. *Global Transitions Proceedings*, *2*(2), 530–534. https://doi.org/10.1016/J.GLTP.2021.08.064

Montevirgen, K. (n.d.). *What are cryptocurrencies and why is the world paying attention?.* Encyclopedia Britannica. Retrieved December 15, 2025, from https://www.britannica.com/money/what-is-cryptocurrency

MURRY CIERRA. (2025, August 28). *Cryptocurrency Explained With Pros and Cons for Investment*. Investopeda. https://www.investopedia.com/terms/c/cryptocurrency.asp