Crypto Chatbot documentation:

Video documentation: https://youtu.be/6kyYixWjkzE (Sorry about the length)

Live Coin Watch (LCW) Data Collection:

In this function, the system uses the URL that has been provided to find key information on the inputted cryptocurrency. To do this the system uses the URL and uses the parameter listed; setting the currency to USD, and code to coin(which the user inputs in the user_input function). The function sets its context type to JSON and uses the API key given to log into our API account. Next, we run the above information to collect the data on the inputted crypto, setting its information provided as 'data'.

After the information has been gathered, we want to know 4 main things, the name, the price, the percent change in the past 24 hours, and the price change in the past 24 hours. To do this, we must look at the documentation of the API to sort through the data the API has provided to find what we want. The API uses a 3D array to store this information. Once we have sorted through and found the desired data, we return the values we have found so that they are stored for when they are called upon.

Coin Market Cap (CMC) Data Collection:

This function works very similarly to the Coin Watch Data Collection function. It similarly begins with using the URL provided to find key information on the inputted cryptocurrency. To do this the system similarly used the URL and the parameters set. In this case, the API key is different and listed outside of the function, so we have to call it into the function in the () when defining the function. Following this, we then run its information above setting its response to 'data'.

Just like the Live Coin Watch Data Collection function, we have to look at the different API documentation to sort through the data. Again, we are looking to collect the same data; name, price, percent change in the past 24 hours, and price change in the past 24 hours. Again, we return desired data to be stored for when they are called upon.

For example, when trying to locate the crypto name, the API goes to data, inputs what the user has made "chat" equal to, and then searches for the name.

User input:

In this function, we begin to present our users. We begin with an introduction to our company. Followed by asking them in they know the symbol of the crypto they want to check, asking them to input 1 If yes, and 2 if no.

Our system then checks their input. If 1 the system will continue, if 2 we will activate 'Prompts'.

Next, we tell the user to input the symbol of the crypto they want to check. Setting their input string as ‘coin’

Our system then takes their ‘coin’ and activates the ‘Live Coin Watch Data Collection’ and ‘Coin Market Cap Data Collection’ functions setting their respective gathered values as ‘LCW_coin_data’ and ‘CMC_coin_data’.

After this, with the newly collected information, we present it to the user telling them the coin they checked, its current price, percent change, and price change in the past 24 hours from both APIs.

Prompts:

In this function, we present the user with examples of a couple of symbols for them to use.

Twitter Data Price

In this function, we are importing the coin data from the two previous API functions. This function is checking if the price of the Crypto on LCW is less than the price on CMC. I have chosen to check if the value is less because then the investor will get the coin at a cheaper price.

So if the price of the Crypto on LCW is less than the price on CMC, the price condition will be set at 1. If this statement is not true, then the price condition will be set as 2.

Twitter Data Percent:

This function is very similar to Twitter Data Price. Here we are importing the coin data from the two previous API functions. Then we are checking to see if the percent change of both cryptos is greater than zero or less than zero. If it is greater than zero we then move on to checking which API has the greatest percentage. If LCW is bigger, the percent condition is set to 1, but if CMC is bigger, the percent condition is set to 2. We then created an ELIF statement at the same indentation as the greater than zero if statement. This is going to check if the percent change from both APIs is less than zero. If this is true then we move onto an if statement seeing if LCW is less than CMC, if this is true, the percent condition is set to 3. But if CMC is smaller than LCW, the percent condition is set to 4.

Alerting User of Twitter Post:

In this function we make the system tell the user that we are going to make a Twitter post using the information they have just found.

Post Tweet:

Inside this function, there are two more functions contained within.

Twitter Post Price

In this function, we are telling the system to check the price condition. It tells the system what to say according to the value of the price condition.

Twitter Post Percent

In this function, we are telling the system to check the percent condition. It tells the system what to say according to the value of the percent condition.

Next, we create an if statement to check if both conditions are greater than zero. If this is true we activate the 'Alerting User of Twitter Post' function. This is followed by the system printing a line to simulate a beginning of a Twitter post. Next, we activate the 'Twitter Data Price' and 'Twitter Data Percent' functions so that we get a new value for both Price and Percent conditions. Next, we activate the 'Twitter Post Price' and 'Twitter Post Percent' functions so the system will print the right text according to the Price and Percent conditions. Finally, we finish off the post with a line simulating an end of a Twitter post.

Main Function:

This is a function where all the magic happens, and everything comes together. This function is set on a loop using a while true statement. It begins with activating the 'User Input' function so that we can see what crypto the user wants to check. Then we set the price condition to 'Twitter Data Price' by activating its function. We also set the percent condition to 'Twitter Data Percent' by activating its function. Finally, we activate the 'Post Tweet' function, where the user will be notified of the Tweet, followed by the simulated Tweet (keep in mind Tweet alert and Tweet will only be posted if the conditions are greater than zero.)

Research on Project Management Approaches:

Review three different project management methodologies (e.g., Agile, Structured, Prototype) and select the one that best suits your project. Justify your choice based on the project's requirements and constraints. Ensure your review contrasts the other project management methodologies (1 page)

For my project, I have chosen to utilise the structured management approach. This is because the structured management approach helps keep clear project objectives, has effective planning, controlled execution, coordinated/ease of task allocation, and ease of documentation.

Clear Project Objectives:

The structured management approach helps clearly provide the objectives and deliverables of the project. The structured management approach looks very similar to the layout of a Gantt chart. Each task has its own set start and due date providing a clear layout of the order each task is going to be completed.

Effective Planning:

The structured management approach also emphasises the importance of project planning. The structured management approach helps define the tasks you want to be done and gives you a timeline of when they need to be completed by.

Ease of allocation:

If this project was to be a collaborative project, the structured management approach is the best option as it provides a way of easily allocating tasks to groups of members. For this project is would have been very easy to allocate different sections of the code for different team members to complete.

Ease of documentation:

The structured management approach is a viable management option for this project as we are required to give in-depth documentation. A structured management approach is the best option for this because once each task is complete we can write documentation based on what we have just completed in this section of the project.

In my opinion, the prototype management approach would not be a useful approach to this project. This is because if we were to give users a limited or partly working project it would be very difficult for them to understand how to adapt the code to do what they require. In addition to this, the project would be very difficult to give to the users at different points during the creation as there is only one final product for them to test at the end of the development. Although user feedback could help incorporate new ideas.

Additionally, the Agile approach would be an okay approach for this project, however, structure is better. This is due to the agile approach being focussed on delivering software fast rather than fully functioning thought-out code. In addition to this, there is less emphasis on documentation for future developers to use and understand what the project is trying to achieve.

These are just a few of the reasons why I think that the structured approach would be best suited to this project over both agile and prototype approaches.

Pseudo Code:

Write pseudo code to outline the key functionality of your app.

IMPORT requests
IMPORT json

Set up our global functions variables;

LCW coin name
LCW coin percent change 24 hours
LCW coin price
LCW coin price change 24 hours
CMC coin name
CMC coin percent change 24 hours
CMC coin price
CMC coin price change 24 hours

Define 'Live Coin Watch Data Collection' (import the 'coin' string):

Bring in our global LCW variables

SET URL to "https://api.livecoinwatch.com/coins/single"

SET payload to json.dumps
SET 'currency' to 'USD'
SET 'code' to 'coin' (string)
SET 'meta' to true

Inside headers
SET 'context type' to 'application/json'
SET 'API key' to 'XXXX-XXX-XXXX-XXX'

SET 'response' to requests.request(USING 'URL' 'headers' and 'payload')

SET 'data' to 'response' from 'json' (running response.json())

PERCENT CHANGE 24h = ROUND the following data (SORT [data] GO TO [0] SEARCH [change] PER [day]) -1) x100) (to get a % value)

LCW NAME = SORT [data] GO TO [0] SEARCH FOR [name]

LCW PRICE = ROUND the following data (SORT [data] GO TO [0] SEARCH [rate]

LCW PRICE CHANGE 24h = ROUND the following data ('PERCENT CHANGE 24h / 100 X 'PRICE')

STORE the values of LCW PERCENT CHANGE 24h, LCW NAME, LCW PRICE, and LCW PRICE CHANGE 24h.

Define ‘Coin Market Cap Data Collection’ (import the ‘coin’ string):

Bring in our global CMC variables

SET URL to "https://pro-api.coinmarketcap.com/v1/cryptocurrency/quotes/latest"

Inside headers
SET ‘accepts’ to ‘application/json’
SET ‘API key’ to CMC API key

SET params to symbol : coin

SET response to requests.get(USING ‘URL’ ‘headers’ ‘params’

PERCENT CHANGE 24h = ROUND the following data (SORT [data] GO TO [quote] SEARCH [percent_change_24h]

NAME = SORT [data] GO TO [coin] SEARCH FOR [name]

PRICE = ROUND the following data (SORT [data] GO TO [coin] then GO TO [quote] SEARCH in [USD] for [price]

PRICE CHANGE 24h = ROUND the following data (‘PERCENT CHANGE 24h / 100 X ‘PRICE’)

STORE the values of PERCENT CHANGE 24h, NAME, PRICE, and PRICE CHANGE 24h.

CMC api key = XXXX-XXX-XXXX-XXX

Define Prompts:
Print/Display “Here are some symbols of crypto's for you to check”

Print/Display”BTC for Bitcoin
ETH for Ethereum
BNB for BNB
MKR for Maker
XMR for Monero
ATOM for Cosmos
LTC for Litecoin
QNT for Quant
AAVE for Aave”

Define User Input
Print/Display “Hi”

Print/Display “Welcome to Zac's Crypto Watch”

SET ‘symbol’ to be an input and Print/Display:
Do you know the symbol of the crypto you would like to check?
1. Yes
2. No"

IF ‘symbol’ is = to 2
Run ‘prompts’

Print/Display “Sure thing, here is all our relevant data on , ’Name’:,

Print/Display “Price of ‘Name
‘LCW Price’ USD from Live Coin Watch
‘CMC Price’ USD from Coin Market Cap

Percent change of ‘Name’ in the past 24 hours:
‘LCW Percent Change 24h’ % from Live Coin Watch
‘CMC Percent Change 24h’ % from Coin Market Cap

Price Change of ‘Name’ in the past 24 hours:
‘LCW Price Change 24h’ USD from Live Coin Watch
‘CMC Price Change 24h’ USD from Coin Market Cap”

Define Twitter Data Price:
Call our global variable of price condition

IF LCW is less than CMC price
SET price condition to 1
Return this value

ELSE:
SET price condition to 2
Return this value

Define Twitter Data Percent:
Call our global variable of percent condition.

IF LCW’s and CMC’s coin percent change 24 hours are greater than 0
IF LCW is greater than CMC’s percent change
SET percent condition to 1
Return this value
ELSE IF LCW is less than CMC’s percent change
SET percent condition to 2
Return this value

ELSE IF LCW's and CMC's coin percent change 24 hours are less than 0
If IF LCW is less than CMC's percent change
SET percent condition to 3
Return this value
ELSE IF LCW is greater than CMC's percent change
SET percent condition to 4
Return this value

Define Alerting User of Twitter Post

Print/Display
"----------
Hi,
Thank you for using Zac's Crpyo Watch
We have noticed that the coin that you have input
has had some significant change recently.

We are just letting you know that we are going to post
this information on our Twitter so that other users like
yourself can receive free investment advice.

For more investment advice, follow Zac's Crypto Watch.

----------"

Define Post Tweet:

Call our global variables;
Price Condition
Percent Condition
Define Twitter Post Price
IF price condition is equal to 1
Print/Display
"'Price of' 'LCW 'Name' is LCW 'Price' USD'
Price was collected from Live Coin Watch
ELSE IF price condition is equal to 2
Print/Display
"'Price of' 'CMC 'Name' is CMC 'Price' USD'
Price was collected from Coin Market Cap
Define Twitter Post Percent
IF percent condition is equal to 1
Print/Display
"'LCW 'Name' has risen by 'LCW Percent'% in the past
24 hours!

This is a 'LCW Price Change' USD increase!

This data was collected from Live Coin Watch

For those that previously invested things are looking good!
Sell when you are happy with profit.

Follow Zac’s Crypto Watch for more posts”

ELSE IF percent condition is equal to 2
Print/Display
“’CMC Name’ has risen by ‘CMC Percent’% in the past 24 hours!

This is a ‘CMC Price Change’ USD increase!

This data was collected from Coin Market Cap

For those that previously invested things are looking good!
Sell when you are happy with profit.

Follow Zac’s Crypto Watch for more posts”

ELSE IF percent condition is equal to 3
Print/Display
“’LCW Name’ has fallen by ‘LCW Percent’% in the last 24 hours!

This is a'LCW Price Change' USD drop!

This data was collected from Live Coin Watch

This is a good time to invest. Buy low sell high.

Follow Zac's Crypto Watch for more posts.”

ELSE IF percent condition is equal to 4
Print/Display
“’CMC Name’ has fallen by ‘CMC Percent’% in the last 24 hours!

This is a 'CMC Price Change' USD drop!

This data was collected from Coin Market Cap

This is a good time to invest. Buy low sell high.

Follow Zac's Crypto Watch for more posts.”

```
RUN ‘Twitter Data Price’
        RUN ‘Twitter Data Percent’

        IF both ‘Price condition’ and ‘Percent condition’ are greater than 0
                RUN ‘Alerting User of Twitter Post’

                Print/Display
                        “----------------------------------------------------------------”
                RUN ‘Twitter Post Price’
                RUN ‘Twitter Post Percent’
                Print/Display
                        “----------------------------------------------------------------”

Define Main:
        LOOP
                RUN ‘User Input’ function
                SET price condition to RUN ‘Twitter Data Price’ function
                SET percent condition to RUN ‘Twitter Data Percent function
                RUN ‘Post Tweet’ function

RUN ‘Main’
```