# SR PROGRAMMING DIVISION CASE

SCHULICH ENGINEERING COMPETITION 2021

As NFT's, Blockchains and Mining Rigs have seemingly become part of the everyday vernacular, your teammates were approached by Elon Tusk to develop a Robust Crypto Portfolio Tracker.

With seemingly every celebrity attempting to cash in on the "create your own crypto" get-rich-quick scheme, Mr. Tusk wants you and your savvy developers to develop a way to track these cryptos and to be able to create a portfolio to track the ones he finds interesting. (And no, it's not insider trading if you jump on his crypto advice, thank de-regulation for that.)

## YOUR GOAL

Your task will require you and your teammates to build and develop a crypto portfolio tracker that utilizes a Crypto tracking API.

For this challenge we recommend Coin Gecko, however there are other API's that will work such as those developed by Coinbase. Please use the one you and your teammates feel the most comfortable with.

## TASKS

a.) The user should be able to create and title a portfolio. This should be constructed through a graphical user interface designed by the team. The more visually appealing and simple and streamlined, the better.

b.) The user should be able to search from a list of coins (that is provided by the API) and then choose to begin tracking those coins within their portfolio. (Crypto Coin Watchlist)

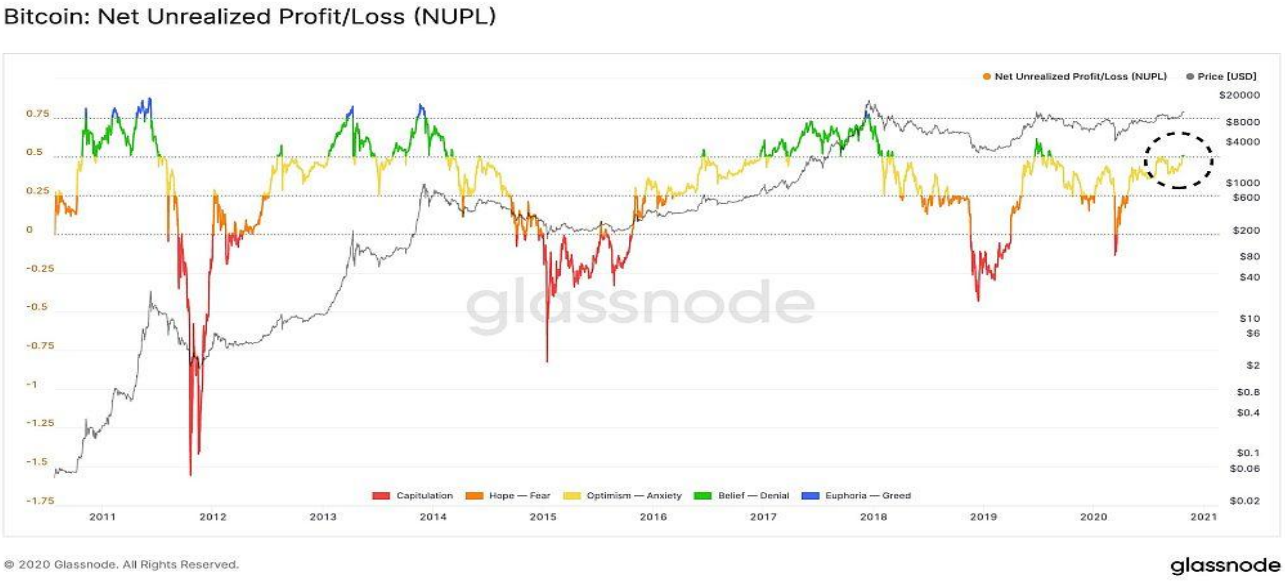
c.) If a person wants to purchase a Crypto, there are two ways this can be done:

1.) They can “quick buy” from their watchlist by selecting the crypto and specifying the amount. If purchased from the watchlist, the Buy date will be assumed to be instantaneous and can’t be modified.
2.) They can access the purchase menu and enter the name of the Crypto, the purchase amount, and the date to be purchased.

d.) The tracker will create a custom profit/loss algorithm to evaluate real quantitative metrics surrounding a Crypto. This profit/loss tracker will receive a Crypto name, the purchase amount. It will then calculate a profit/loss based on the instantaneous value. (Bonus marks: if you can develop a robust algorithm to predict profit/loss based on extrapolated “future” values of a specific crypto)

Profit/Loss Visualization

The user should select a range (i.e. start date and end date) starting with their buy price, and they should see a period (visualization - a graph ) of profit/loss for that asset. They should be able to see this for different date ranges.




You can utilize any viable Crypto API to visualize profit/loss.

e.) Compare asset performance - For each asset, the tracker should be able to compare the performance of each asset from their buy date to the present, with the performance of major crypto such as Bitcoin and Ethereum during that date range. This comparison should be visualized as well.

You can utilize any viable Crypto API to visualize asset performance.

f.) Visualize total portfolio performance - Once again you should be able to visualize your total portfolio performance, while giving the user the ability to toggle which cryptos they want to visualize in their portfolio.

Visualization includes:

a.) Date Range
b.) Crypto Toggle
c.) Profits over time

You can utilize any viable Crypto API to visualize asset performance if you desire.

g.) Last but definitely not least, Mr. Tusk wants you to create a robust risk analysis evaluation tool. (Strategy/Algorithm)

When the portfolio is passed through the evaluation tool the GUI must provide a conclusion:

1. Very High Risk
2. High Risk
3. Neutral
4. Low Risk
5. Very Low Risk

Your team will be able to develop this algorithm in whatever way you choose. However the more robust and nuanced your algorithm is, the higher points you will score. (Remember, millions of dollars are potentially at risk!)

Some factors you can evaluate:

- Marketcap (lower marketcaps < $100M are considered risky)
- Recent increase in price (over 100% increase in one week makes a coin risky to hold as it may experience a correction)
- Other factors?

# Presentation

Once you have completed your crypto tracker, each group will give the judges a demo/presentation as a pitch for your tracker to Elon Tusk's representatives (The Judges). It should outline how your product meets or exceeds the expectations and why it is the best choice. You will have approximately 10 minutes for the presentation/demo followed by a 5 minute question period from the judges over Zoom. See the table below for evaluation metrics.

Order of presentations will be randomly chosen and communicated to teams 30 minutes prior to the end of the work time. Submission details will be specifically explained by the division's Director during the briefing period.

Recognize that the project outline is the base case solution. Adding more to your project is recommended for a better scoring solution. However, please attempt to fulfil the requirements outlined above before venturing outward into adding more functionality.

# Submission

Before 1:45 you must email a github link to your project repository to amneet.deol@ucalgary.ca. Your subject must include: Sr. Programming Division Submission - (Last Names of each team member)

Failure to submit by 1:45 will disqualify you from the round.

# Evaluation Metrics

| Programming Judging Matrix | | |
|---|---|---|
| Strategy/Algorithm | Simplicity | /10 |
| | Ingenuity | /10 |
| | Ability to Achieve Desired Outcome | /15 |
| | | **/35** |
| Code | Structure | /10 |
| | Consistency | /5 |
| | Readability | /10 |
| | Efficiency | /10 |
| | | **/35** |
| Resource Management | Memory Usage Efficiency | /5 |
| | Program's CPU Usage | /5 |
| | | **/10** |
| Presentation | Design Process and Justification | /7 |
| | Design Critique | /4 |
| | Voice, Articulation and Timing | /4 |
| | Visual Aids | /2 |
| | Response to Questions | /3 |
| | | **/20** |
| Deduction Total | | |
| | | |
| **Total** | | **/100** |