**Sprint 1**

1. Pick 1 instrument from all coin pairs(ADAUSDT) – in future, user should be able to enter watch list for which to trade on with any number of coins
2. Check for condition where (Close(0)/Close(1)) <0.9
3. Initial Account Value is $1000
4. Maximum position is 1( we are using only ADAUSDT)
5. If Condition in Step 2 is true, buy $1000/Close price of ADAUSDT @ Close[0]
6. Set Maximum Trade Duration from Close[0] day, default 3 days
   1. If after 3 days from trade profit is not met, exit at Close[0] price of Time(Day 3 after)
7. Set Profit Target of X = 10%
   1. If Close[0] is 10, then at each Close of each bar less than Maximum Trade Duration days, you check if Close[0]>=Profit Target.
   2. If yes, exit at Close[0] which is greater than or equal to profit target
8. After each trade, a new account balance is calculated based on Profit or loss and that is used to divide over the Close[0] price for subsequent trades
9. Allow user to enter Start and End time for back test
10. At end of backtest, exit all positions at Close[Tfinal]
11. Report all trades entered and exited
    1. Trade #|Crypto|Entry Date|Entry Price|Exit Date| Exit Price|Coins Purchased|Exit Signal(Backtest End or Profit or Duration Exit)
12. Report Performance metrics
    1. Profitable Trades = (Total trades with gains/Total Trades)\*100
    2. Final Account Balance=Initial Account value+Losses+Gains
    3. Profit Factor= Total Gain/Total Losses
    4. Total Trades
    5. Gain Percent = (Final Account Value-Initial Account Value/Initial Account Value)\*100

**Sprint 2**

1. Pick 3 instrument from all coin pairs(ADAUSDT,ETHUSDT &ETCUSDT)
2. Check for condition where (Close(0)/Close(1)) <0.9
3. Initial Account Value is $1000
4. Maximum position is 3( we are using only ADAUSDT,ETHUSDT &ETCUSDT)
5. If Condition in Step 2 is true, buy $1000/Close price of ADAUSDT or ETHUSDT or ETCUSDT @ Close[0]
6. Set Maximum Trade Duration from Close[0] day, default 3 days
   1. If after 3 days from trade profit is not met, exit at Close[0] price of Time(Day 3 after)
7. Set Profit Target of X = 10%
   1. If Close[0] is 10, then at each Close of each bar less than Maximum Trade Duration days, you check if Close[0]>=Profit Target.
   2. If yes, exit at Close[0] which is greater than or equal to profit target
8. After each trade, a new account balance is calculated based on Profit or loss and that is used to divide over the Close[0] price for subsequent trades. So if we made 300 on prior trade, we use 1300 and divide by 3 for each next position. As trades are exited, a new account balance is calculated and divided over the positions for trading
9. Allow user to enter Start and End time for back test
10. At end of backtest, exit all positions at Close[Tfinal]
11. Report all trades entered and exited
    1. Trade #|Crypto|Entry Date|Entry Price|Exit Date| Exit Price|Coins Purchased|Exit Signal(Backtest End or Profit or Duration Exit)
12. Report Performance metrics
    1. Profitable Trades = (Total trades with gains/Total Trades)\*100
    2. Final Account Balance=Initial Account value+Losses+Gains
    3. Profit Factor= Total Gain/Total Losses
    4. Total Trades
    5. Gain Percent = (Final Account Value-Initial Account Value/Initial Account Value)\*100

**Sprint 2**

1. Assume all USDT instruments are your watch list and trade them
2. Check for condition where (Close(0)/Close(1)) <0.9
3. Pick the Cryptos with the lowest 10 of Step value and invest in them
4. Initial Account Value is $1000
5. Maximum position is 10( we are using only ADAUSDT,ETHUSDT &ETCUSDT)
6. If Condition in Step 2 is true, buy $1000/Close price of any of the 10 cryptos in ascending order @ Close[0]
7. Set Maximum Trade Duration from Close[0] day, default 3 days
   1. If after 3 days from trade profit is not met, exit at Close[0] price of Time(Day 3 after)
8. Set Profit Target of X = 10%
   1. If Close[0] is 10, then at each Close of each bar less than Maximum Trade Duration days, you check if Close[0]>=Profit Target.
   2. If yes, exit at Close[0] which is greater than or equal to profit target
9. After each trade, a new account balance is calculated based on Profit or loss and that is used to divide over the Close[0] price for subsequent trades. So if we made 300 on prior trade, we use 1300 and divide by 3 for each next position. As trades are exited, a new account balance is calculated and divided over the positions for trading
10. Allow user to enter Start and End time for back test
11. At end of backtest, exit all positions at Close[Tfinal]
12. Report all trades entered and exited
    1. Trade #|Crypto|Entry Date|Entry Price|Exit Date| Exit Price|Coins Purchased|Exit Signal(Backtest End or Profit or Duration Exit)
13. Report Performance metrics
    1. Profitable Trades = (Total trades with gains/Total Trades)\*100
    2. Final Account Balance=Initial Account value+Losses+Gains
    3. Profit Factor= Total Gain/Total Losses
    4. Total Trades
    5. Gain Percent = (Final Account Value-Initial Account Value/Initial Account Value)\*100

**Note: If a position is exited, we calculate the value for step 2 and enter the crypto out of the 10 with the lowest for that time step if a trade can be entered. If 2 positions are open, we enter with the 2 lowest values of step 2 etc**

**4. Sprint 4**

1. Allow user to repeat Sprint 3 with Different Watch lists
2. Allow user to perform multi sorting inorder to trade
   1. Duration: 5,10,20 Days Profit Target: 10,20

Strategy 1

Step 1: Close[0]/Close[1]<0.9 and Sortby(Lowest 20, Close[0]/Close[1])

Step 2: enter 10 lowest (Lowest 10, Close[0]/Close[1])

Strategy 2

Step 1: Close[0]/Close[1]<0.9 and Sortby(Lowest 20, Close[0]/Close[1])

Step 2: enter 10 lowest ((Close[0] - Close[2])/Close[2])

Strategy 3

Step 1: Sortby(Lowest 10, (Close[0] - Close[2])/Close[2])

Step 2: enter 4 lowest ((Close[0] - Close[14])/Close[14])

Strategy 4

Step 1: Sortby(Lowest 10, (Close[0] - Close[14])/Close[14])

Step 2: enter 4 lowest (Close[0])/Close[1])

Strategy 5

Step 1: Sortby(Highest 10 ATR 2)

Step 2: enter 4 lowest (Close[0])/Close[1])

Strategy 6

Step 1: Sortby(Lowest NATR(2)) Highest 10

Step 2: enter 4 lowest (Close[0])/Close[5])

Strategy 7

Step 1: Sortby(Lowest 10, (Close[0] - Close[14])/Close[14])

Step 2: enter 4 lowest (Close[0])/Close[5])

Strategy 8

Step 1: Close[0]/Close[5]<0.9 and Sortby(Lowest 10, Close[0]/Close[5])

Step 2: enter 4 lowest (Close[0] - Close[2])/Close[2])

Strategy 9

Step 1: Close[0]/Close[5]<0.9 and Sortby(Lowest 10, Close[0]/Close[5])

Step 2: enter 4 lowest (Close[0] - Close[14])/Close[14])