|  |
| --- |
| Crypto Books  CSCI 230 Final Project |
| Jasper Heiste |



## Crypto Books

## Project Summary

In the crypto currency space, there is not a really good way at this point to keep track of all of your different investments across different platforms. The idea for Crypto Books is to be able to compile these stats and keep track of gains and losses and account sizes. The first implementation will be tracking bitcoin,litecoin, ripple, and ethereum. Then add more coin support possibly. The intended user is somebody who has crypto currency interests and has a good knowledge of crypto currency lingo. The problem is a lack of a book keeper for your own crypto currency. I would like to make a web app with HTML pages. I would like to display info and real time calculations by using an api from an exchange website. I will also need to save info as well preferably using .csv files. The .csv file implementation is not really started yet but the primary way was entiering them manually and that is implemented on the edit transactions (append.html) page.

### **Use Case Analysis**

### There will be three main ways that the user will interact. The can enter sells or buys, upload a document of sales and buys, or view sales and buys.

### Enter sales and buys

### The user will enter the date, the amount bought (in USD) of the coin they selected. They will then select whether it was a sale or a buy.

### Upload a document of sales and buys (not completed. I look to add support for this In the future. I opted for other features to have done)

### Many exchanges allow you to download your history in a .xlsx file. I will use this by converting it to a .csv file. I will integrate support from one of the biggest exchange, Binance.

### The user will click upload and route this file so I can manipulate it in the program.

### View sales or buys/profit (history page/dashboard page)

### History-The user will be able to navigate to a page that displays prints all transactions in a table

### Dashboard-in a table shows holdings of a selected coin, average cost, current price , and the difference in percentage gained or lost. Using python and in line html style, I make the percentage red if it is a loss of value and green if it is a gain of value

Data Design

The data will be kept using the mysql database provided by pythonanywhere and then get a current value for the amount of bitcoin, litecoin, ripple, or ethereum held I use the **mysql.connector to do this and put the data into a tuple** which I commonly call histlist. I understand after the fact how misleading this is but I did not want to go through and change everything to risk messing it up. After that, I will compare the value of what is held to the initial value of USD used to purchase the coins. I use the coin marketcap.com api to access current prices. **Using the ‘requests’ library I am able to get the info and I read it with the ‘json’ library.**

The dates are going to be important because I would like to be able to find the price of the coin at a certain price on a certain date. There will be no other manipulation of this. The data will need to be converted into a form that I can compare with data from online as well as a form for the user to view when looking at history.

Algorithm

* Dashboard
  + Access api form coinmarketcap.com to get price of targeted coin as determined from post method in dropdown menu of this page that posts to itself
  + Using formulas like the difference formula and percent change formula as well as other algebraic equations, get the total amount of that type of coin owned by referring to the transactions table in the database
* History page class
  + Needs to print history of transactions and total at that point as well as gains and losses
  + Needs a button function
* Add Sales Page
  + Enter transactions manually and have alink to delete page
* Delete page
  + Shows all transactons with a delete button by each one. The delete button stores a value which is the id number in the mysql table used on the next page to confirm the deletion and then delete the transaction

# UI Design (see below)

# Adding(with link to deletion page)

# Deletion page

# Displaying stats (history page)

# Dashboard page to display real time data and claculations

# Design

# Slate color with black text

# Gradient slate on the background

# Boxes a darker grey almost black with black text

# Orange accent in menu, buttons and footer

# Dashboard

# ../Desktop/Screen%20Shot%202018-04-23%20at%202.26.29%20PM.png

# History

# ../Desktop/Screen%20Shot%202018-04-23%20at%202.26.37%20PM.png

# Append (incliding a page that reports that the transaction has been entered)

# ../Desktop/Screen%20Shot%202018-04-23%20at%202.26.44%20PM.png

# Delete ../Desktop/Screen%20Shot%202018-04-23%20at%202.26.55%20PM.png

# Verify deletion (page that reports delete as successful)

# ../Desktop/Screen%20Shot%202018-04-23%20at%202.28.50%20PM.png