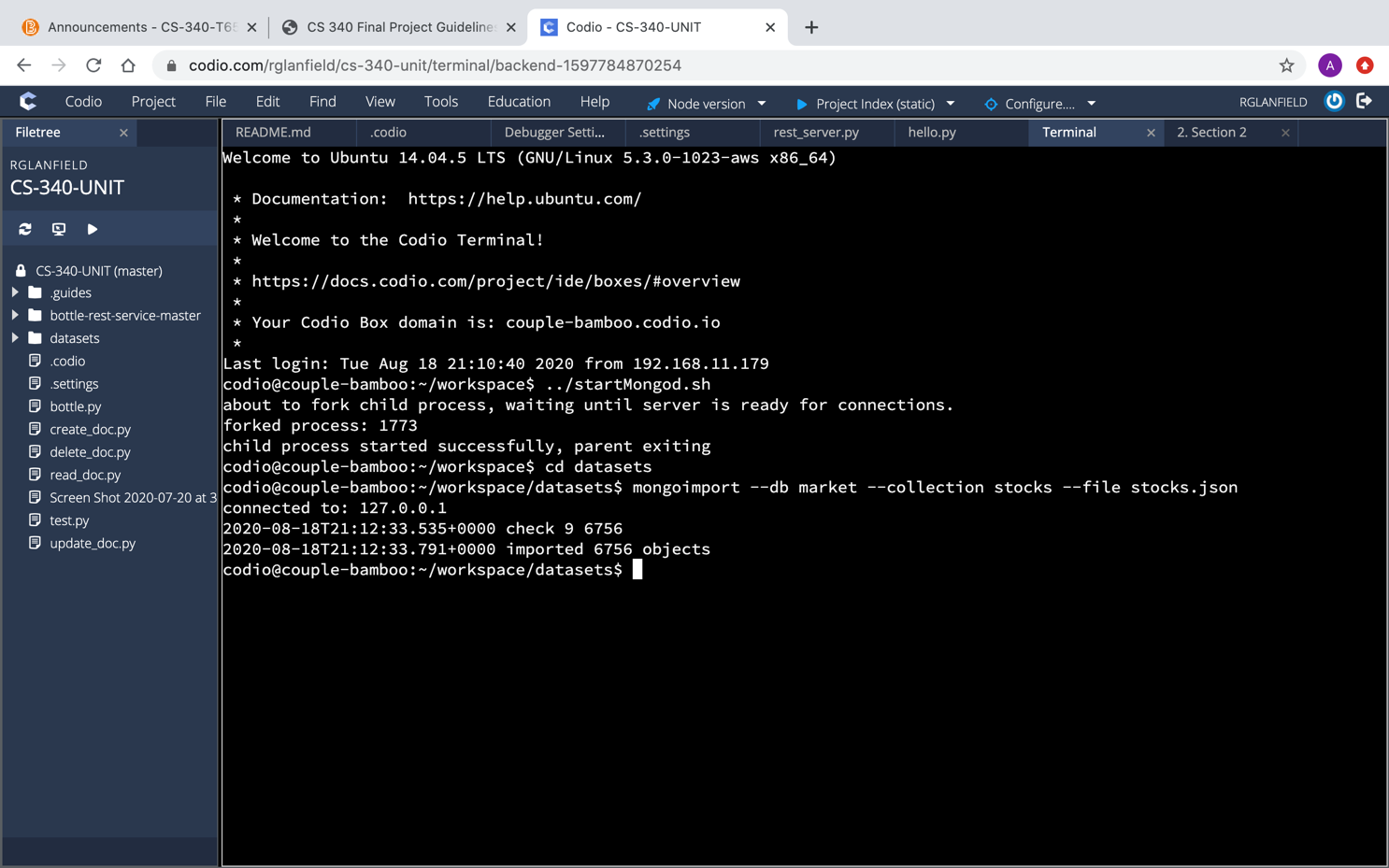
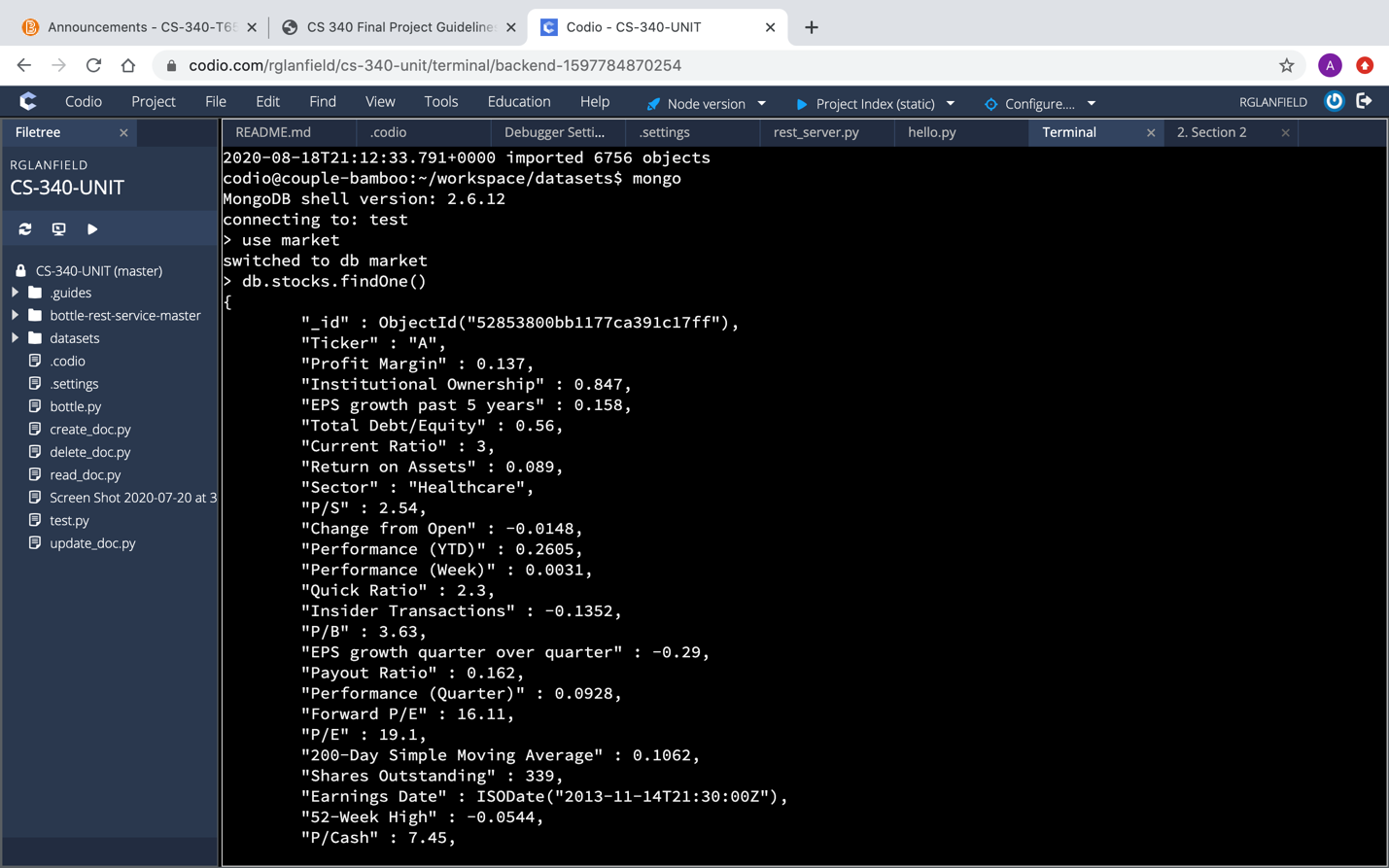
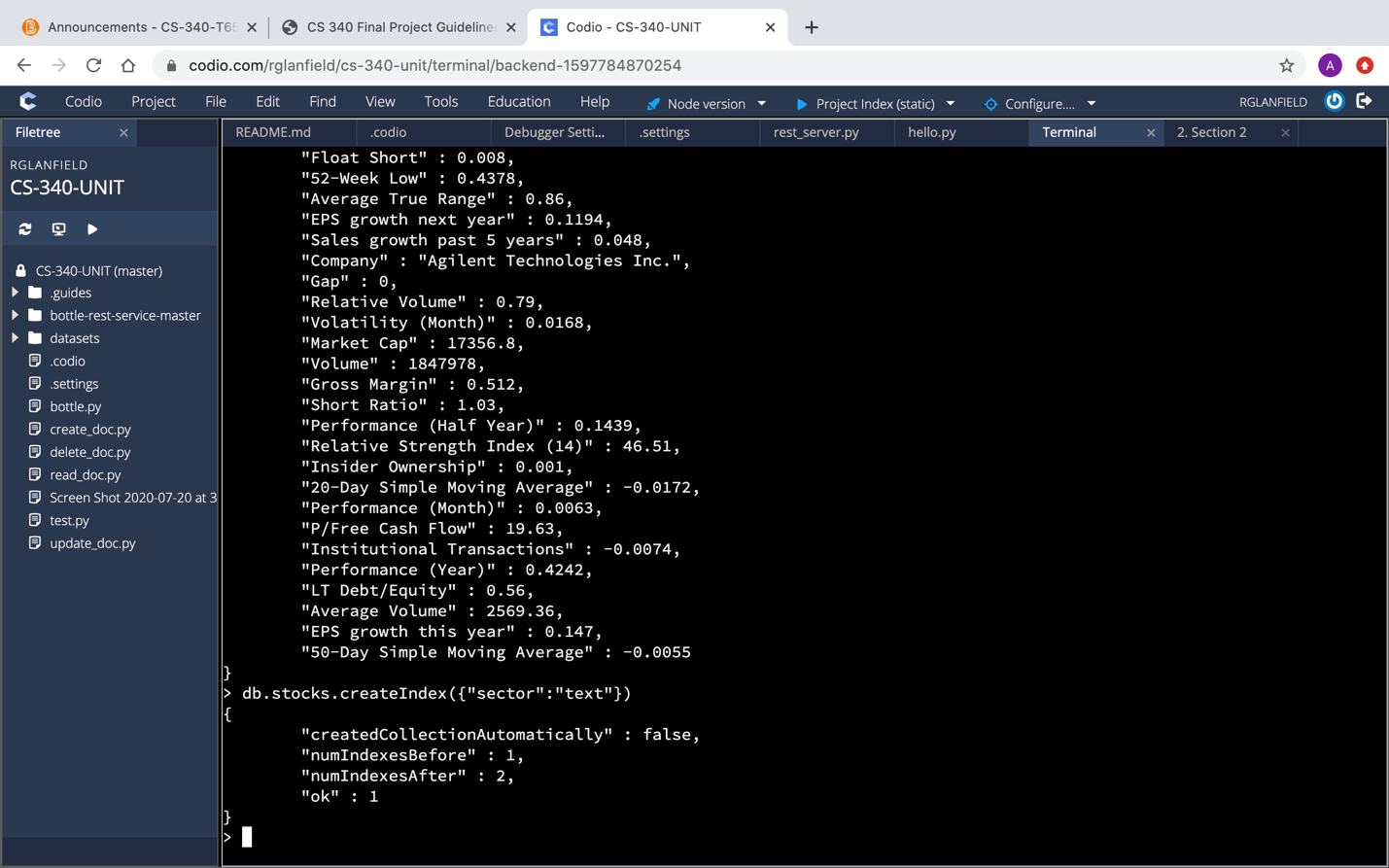
I.

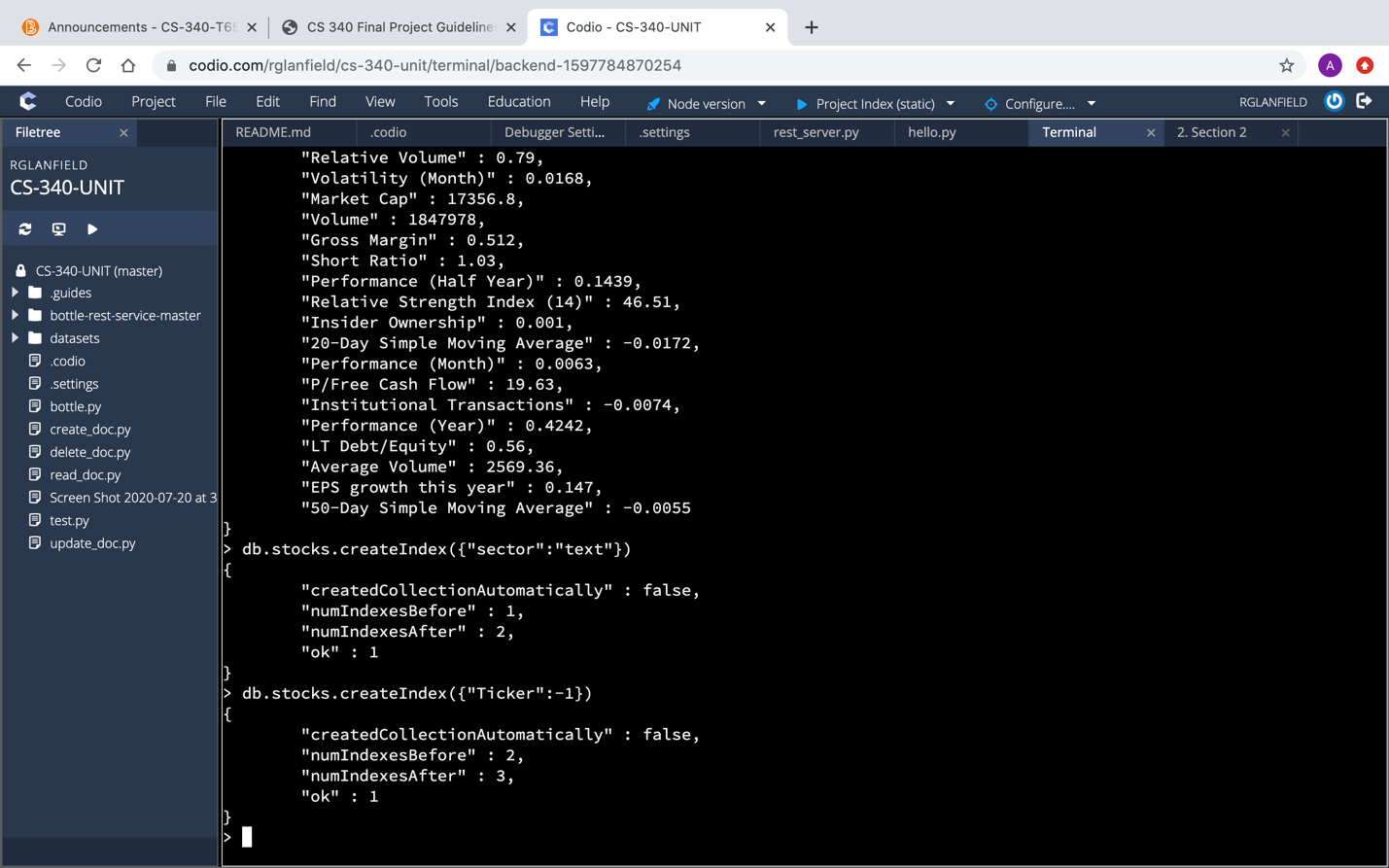
A. I utilized the mongoimport tool to create a database named market and a collection named stocks.This database loaded with documents from the stocks.json file. Below, you can see the command returned the result; imported 6756 objects. This verifies the files were imported into the created database and collection. I also went in to the market database I created and further verified the files were in the stocks collection using the command db.stocks.findOne().





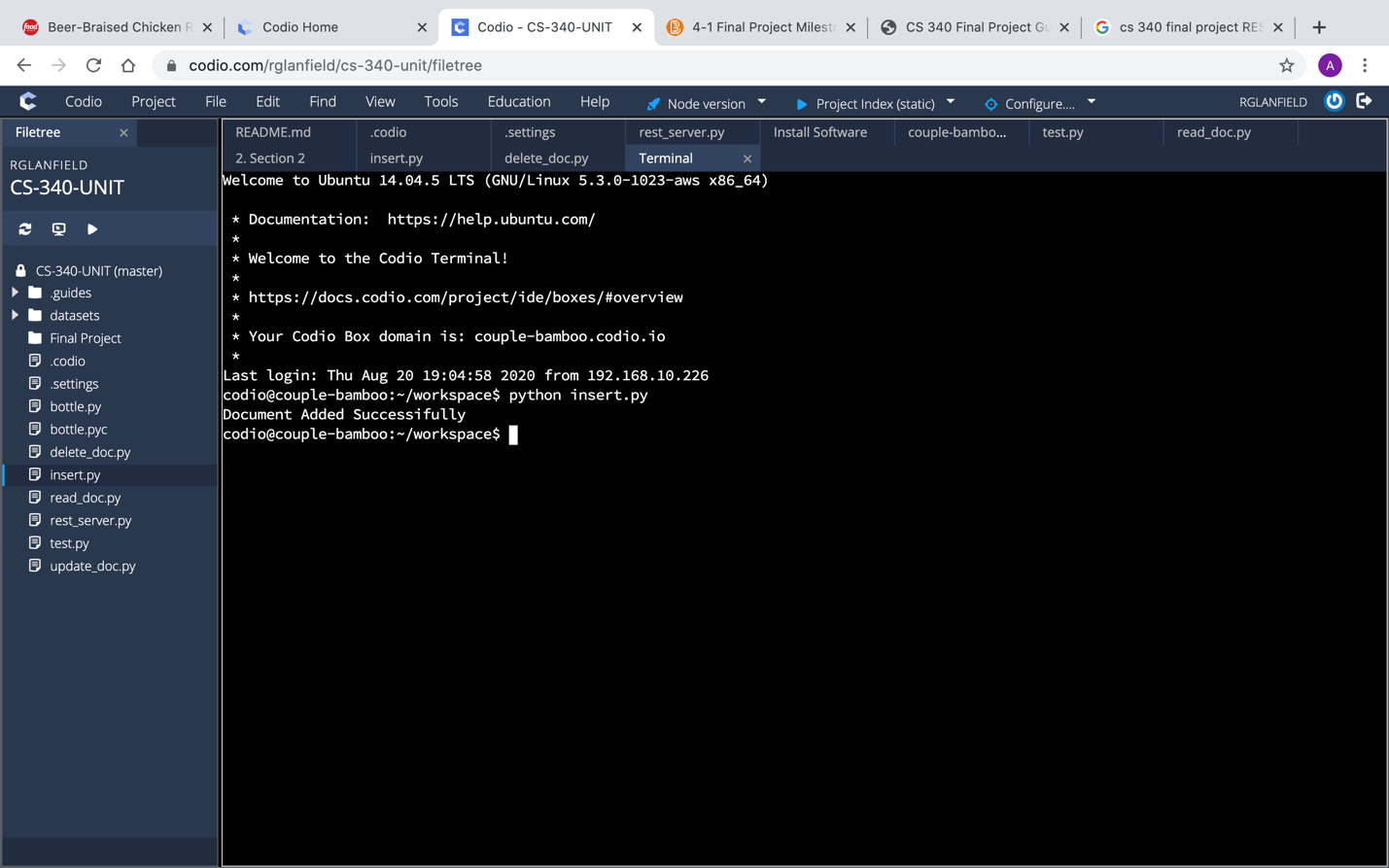
B. Indexes were created to improve searching for documets through the database. It enables faster searching by looking through the indexes for documents with specific keys rather than every single document. Both single and compound indexes were created, as shown below.



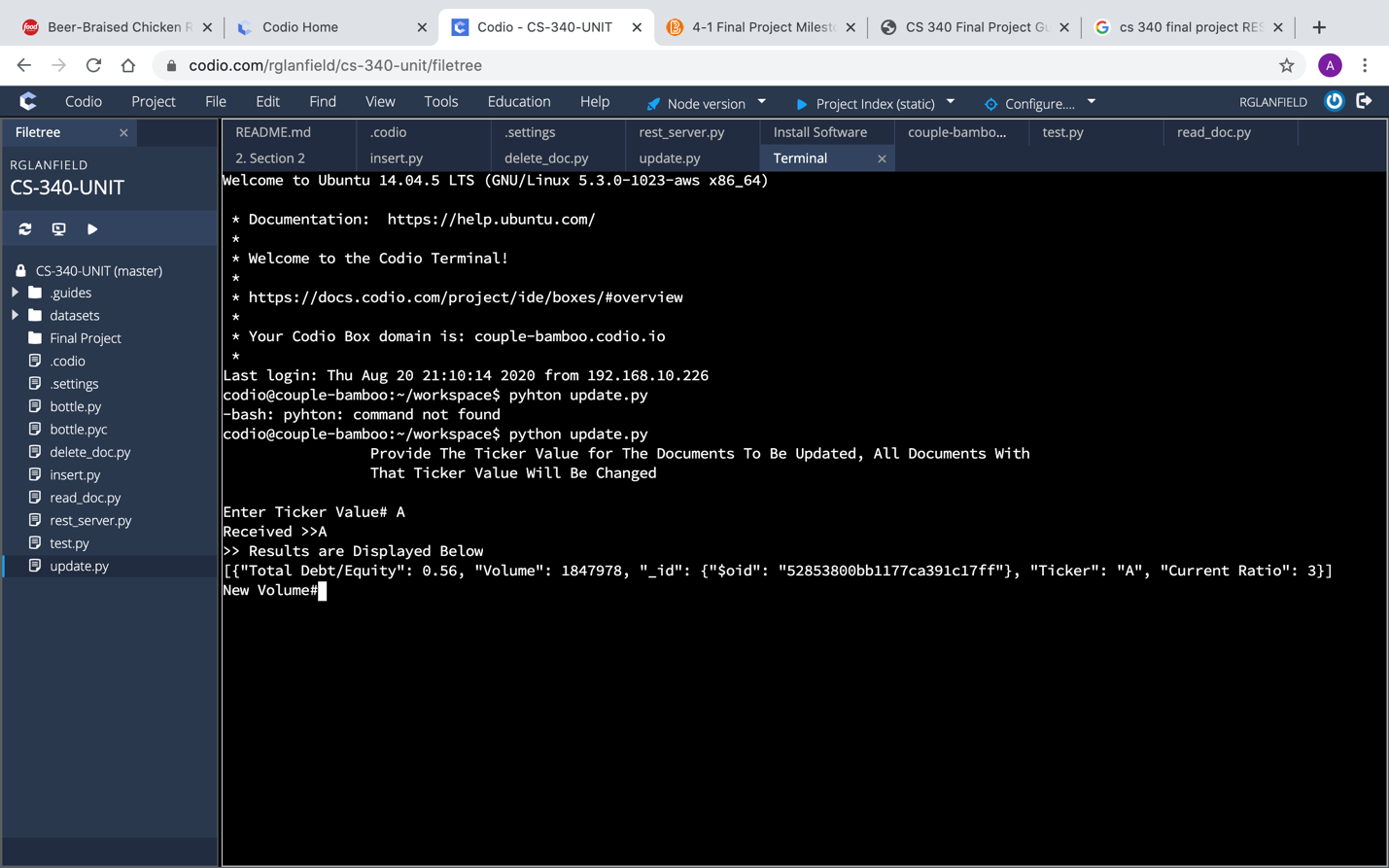


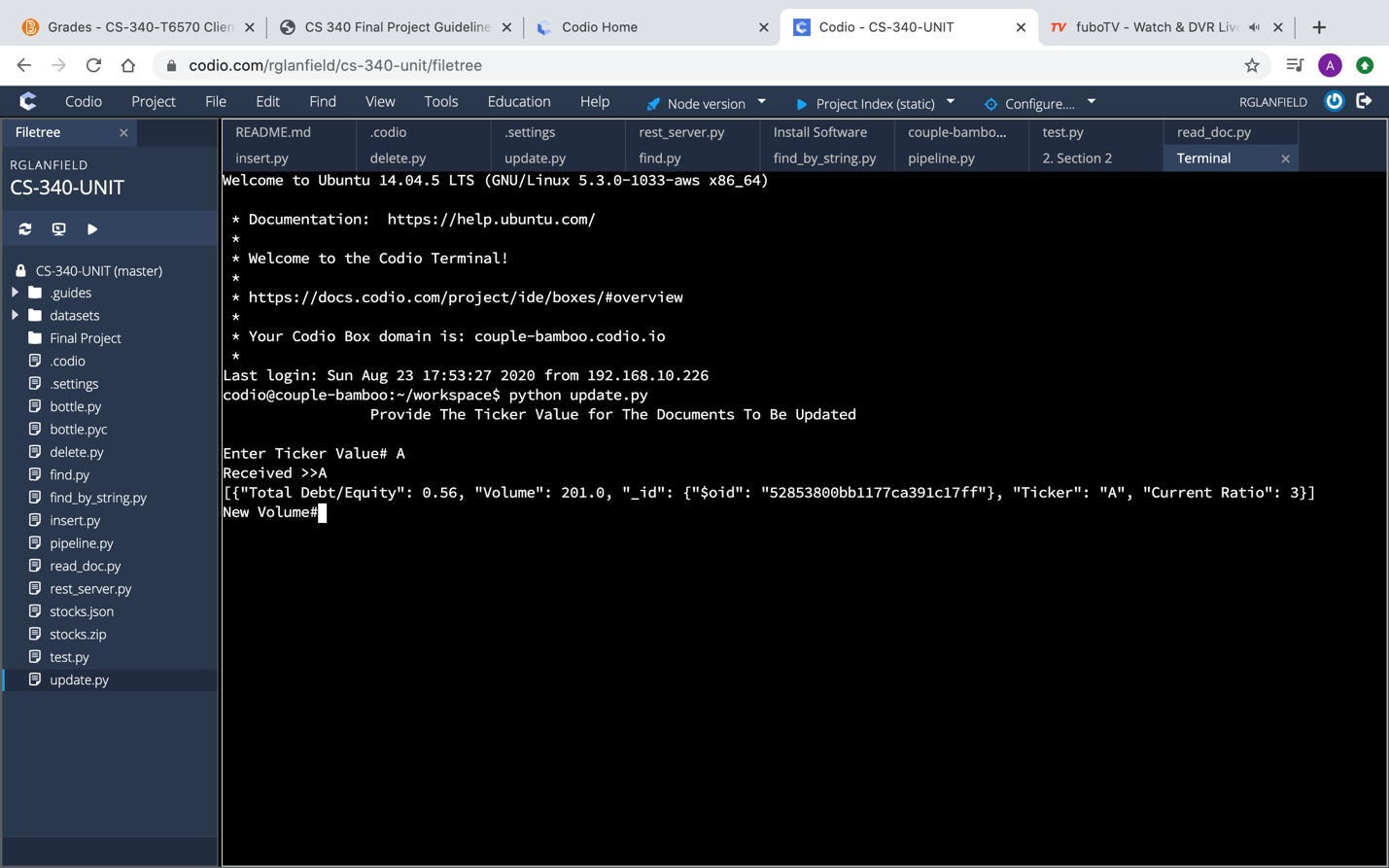
II.

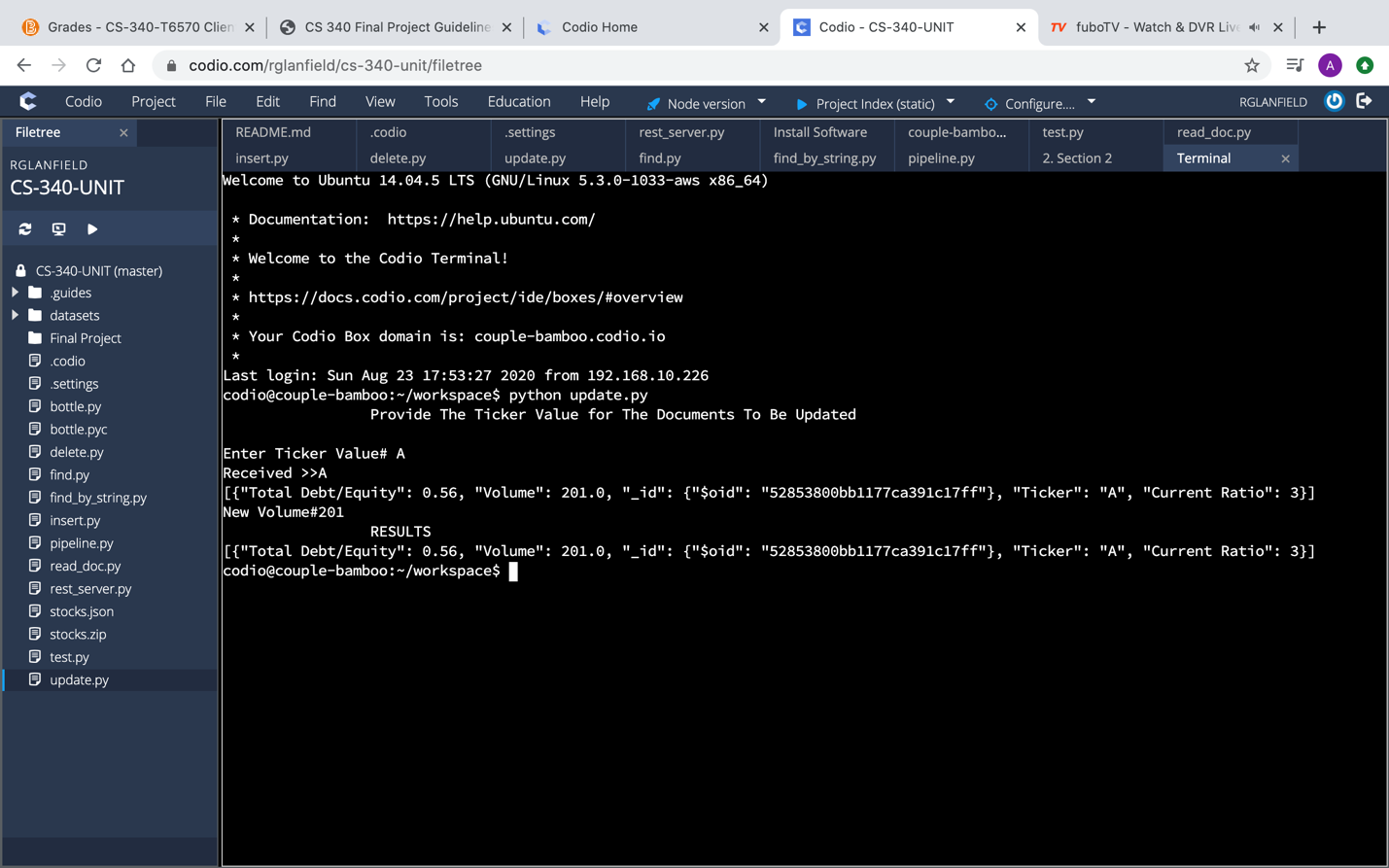
A. Using the Python programming language I was able to insert new key-value pairs into documents in the created collection. The function created reads from a file a value pair stream in JSON notation and inserts this document into the stocks collection. In addition, I created an application scaffold for testing purposes.



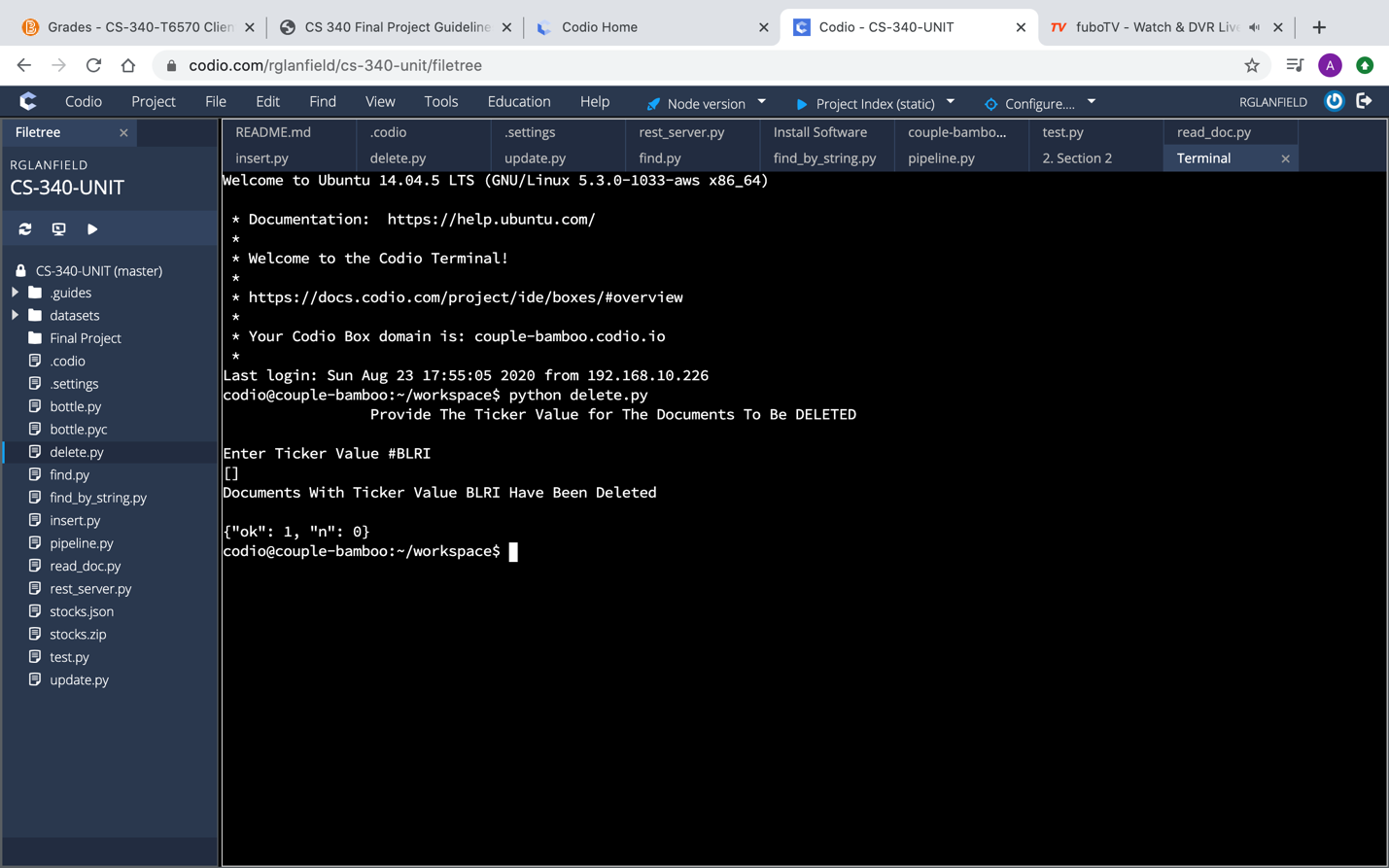
B. Next, I created a function that will allow for MongoDB statements to be used to update the document "Volume" key-value pair identified by the string input stock ticker symbol; "Ticker", and numerical input; "Volume". This includes a main application to call the function.







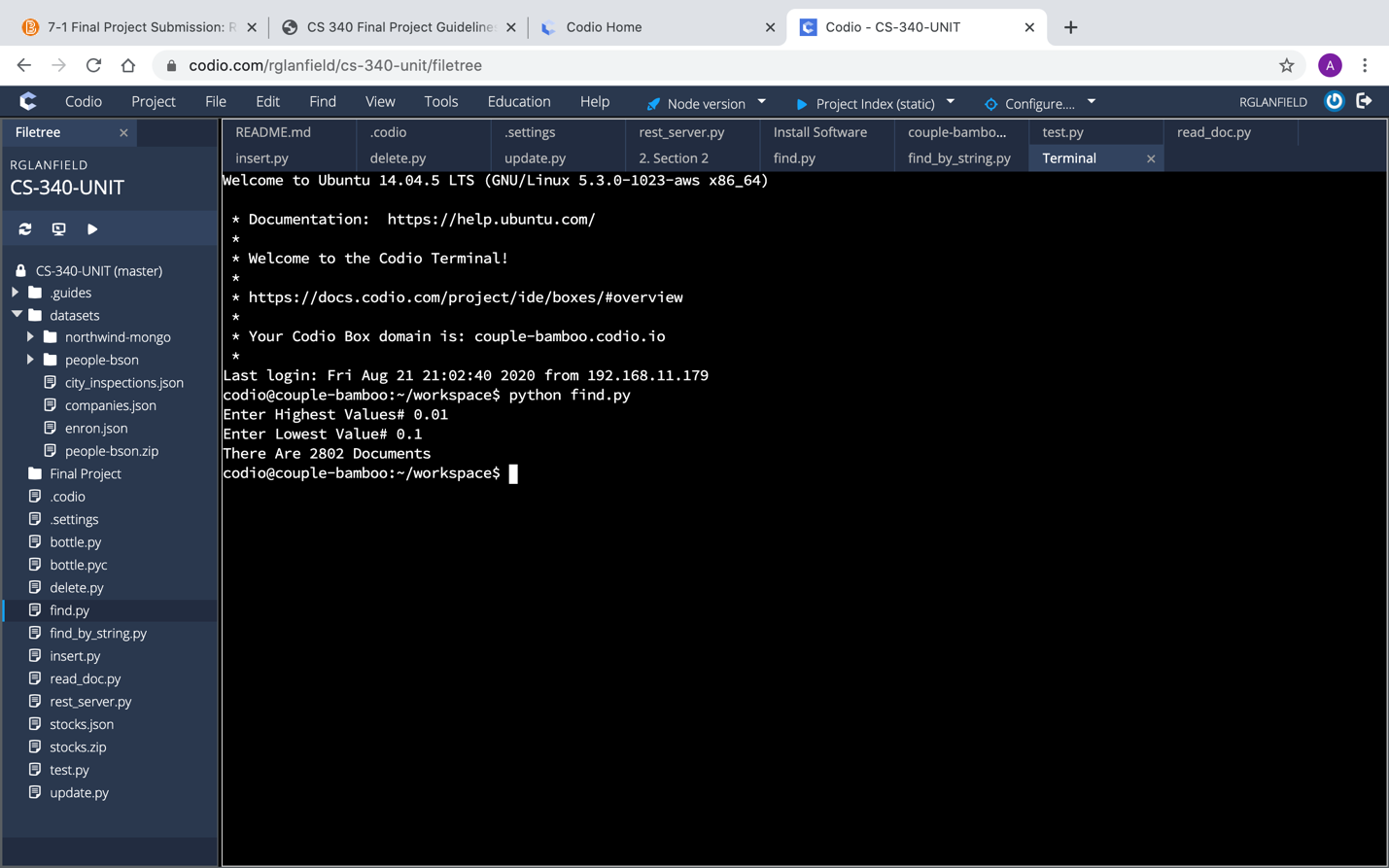
C. A function was created to enable existing documents to be deleted using MongoDB statements. The function takes ticker symbol "Ticker" input and removes the document identified by the given ticker symbol. I also created a simple application scaffold to test the function.

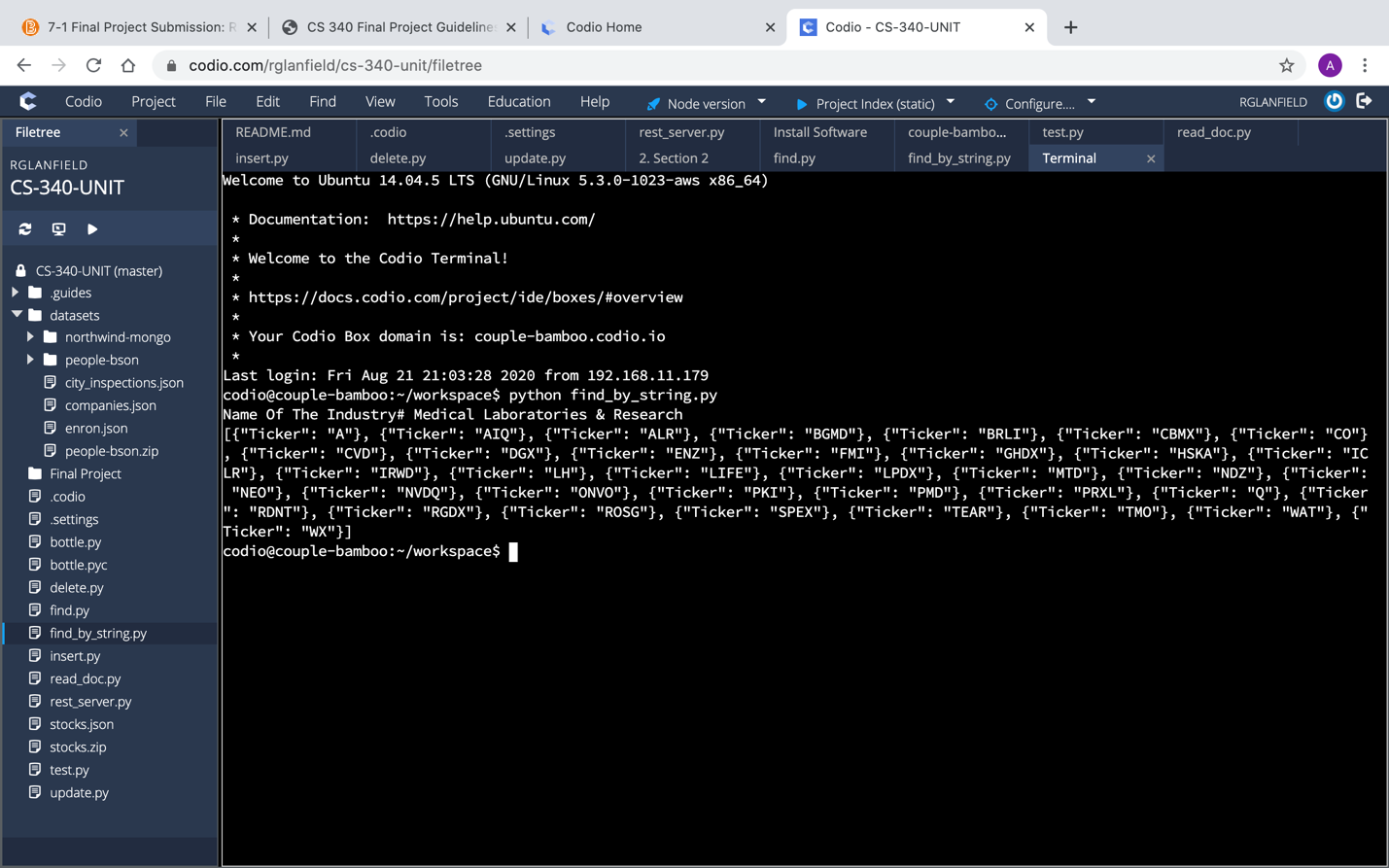


III.

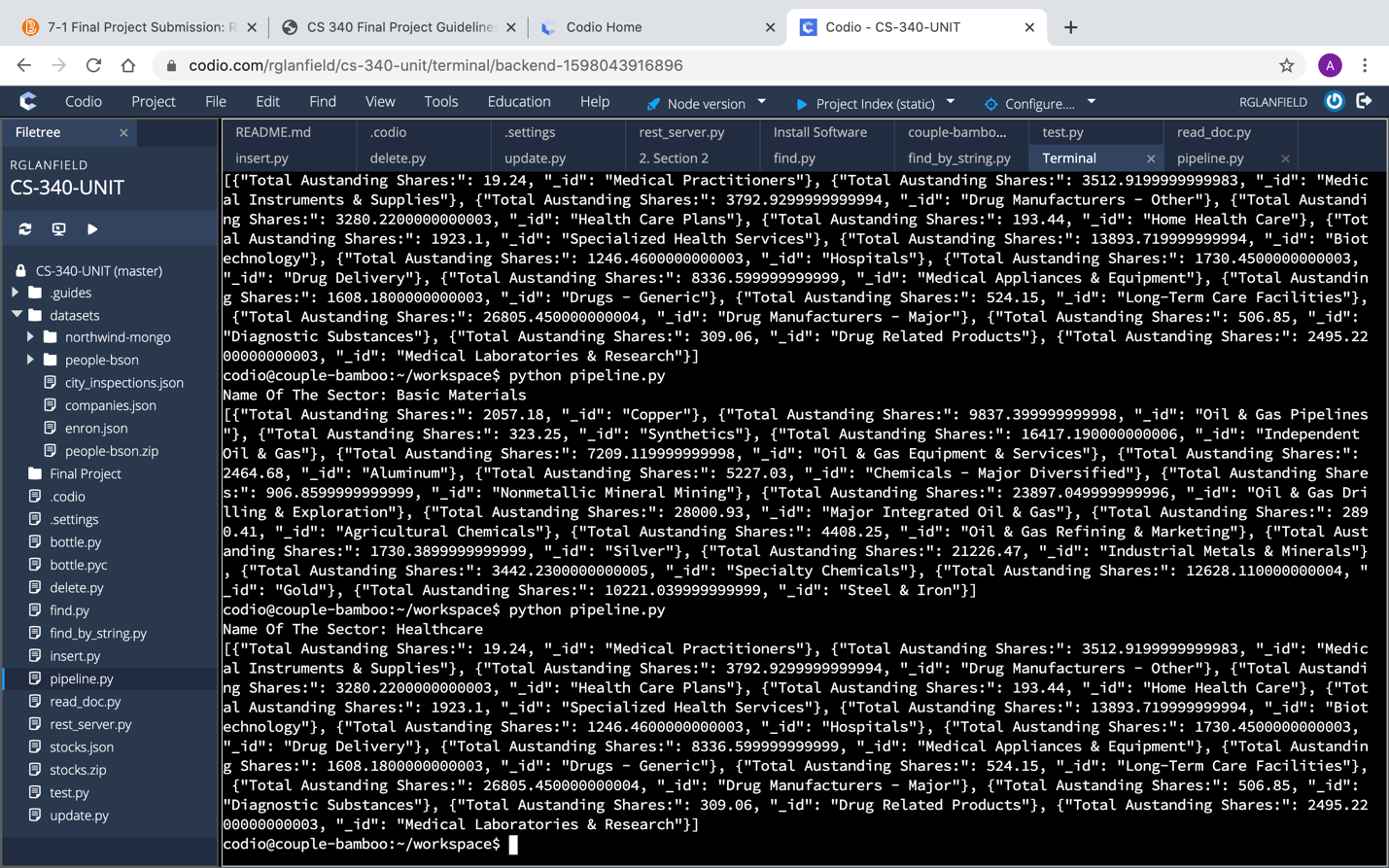
A. A function was also included for retrieving documents. The function takes input for low and high numerical values. Below, you can see that it finds documents for which the "50-Day Simple Moving Average" is between the low and high values entered and returns the count

of the number of documents found. I also included a function that will take as input a string. The function or method will find documents for which the input string matches the document key "Industry" and returns the list of ticker symbols found to match that industry. You can see the results of my search using the industry string, "Medical Laboratories & Research".





B. There is also a funtion that allows a user to write MongoDB aggregation pipeline statements that transforms documents into aggregated results. The function takes string as input and finds documents for which the input string matches the document key "Sector" and returns the total outstanding shares grouped by document key “Industry.” Shown below are the results returned using sector string inputs,"Healthcare," and "Basic Materials".



IV.

A. I also developed a web service application to implement a RESTful application programming interface (API) for a MongoDB database. I developed it using the Python web service framework for the collection of stock market summary data. The RESTful API allows the user to create, search, update and delete documents.

