Business Application Development
Dr. Castillo

Tech Stocks
Gui-Enabled Stock Miner

—

By Carlos Bonilla, Mikayla Bartell, Alex Huerta

The Debut.

From implementing the essentials of Java Programming to using its distinctive capabilities, we introduce a Graphical User Interface that showcases the Stock evaluations of the impactful Silicon Valley tech companies of Apple, Microsoft, Amazon, Intel, and Google. Through their current and historical stock prices, we analyzed their spring quarter evaluation by using integrated tables and graphs.

The Story Behind The Idea

Now a days, one can easily just ask a home assistant or just search online to find out the days stock price, but why not just create a program that gives you all the historical data of a stock at the click of a button. With that mindset of thinking outside the box, our team went back and forth on how we would be able to create a user-friendly program to analyze a company's stock data. After brainstorming for days, the team pondered on the conclusion of trying to achieve a basic interface with back end programming code that will be able to web scrape a company's historical stock data. Not being computer science students, our team wanted to go above and beyond and create an algorithmic interface that will coincide with the Orfalea College of Business.

How to Use the Stock Miner Graphical User Interface

1. Download the "Stock Miner" Graphical User Interface and Related Files

User will download the Program code and related files from Alex Huerta's PolyLearn Group Project Submission. Within the submission the user will be able to the find Documentation regarding the project, Stock Database, StockMiner.Rar file, and the Source code for the interface.

2. Move corresponding "Stock Miner" Files to User-Designated Folders

Once all the related files have been downloaded by the user, one will have to move the database file into the user's desktop and/or specified computer file location. The user will be using the stock database in the program, so they will have to know where it is currently stored.

3. Open up the "Stock Miner" Interface with the .JAR File

Open up the Graphical-User Interface corresponding to the "Stock Miner" JAR file. Depending on your computer (Mac or PC), the user will double click the JAR file or right click on it and selected "run". Once the user has done so, they will be able to get to the opening of a blank interface.

4. Click Open Database

5. Browse the File Explorer to Find the Location of test.db

6. Click on the Combo Box to Select the Stock Desired From the List

Once the user clicks on the ComboBox it will let one select AAPL, GOOGL, AMZN, INTC, and to see its desired historical and current data related to its stock.

7. Select the Stock

Once the user selects one of the stocks from the combo box, the table will populate with the selected stock's data such as date, open and close price, volume, and the days price change percentage. Also when the stock is selected from the combobox, it will auto pop-up a multi chart regarding the closing prices throughout the spring quarter. When the user selects another stock from the combo box it we redo all the automatic steps just stated previously.

Course Elements Used

Methods

The Stock Miner team incorporated methods from Professors "SandBox" class and as well programmed their own. Those methods include collecting Stock Data(which holds statements regarding the web scrape), editing the createMultiLineChart method, and changing the way of the stock data was being populated into the JComboBox.

Loops

In regards to implementing loops into our Stock Miner interface, the team designed a "while" and a "for" loop to generate an integer to store it into the Stock database. Being that the data we web scraped was downloaded as a set of strings into a text file, the team programmed a loop to convert the "date" of the stock's closing prices to an integer so the chart methods can implement it's desired values.

Gui's

In order to achieve a user-friendly interface to view the stock data that the Stock Miner team web scraped, we created a simple GUI. Within the GUI, the user will be able to select one of the tech stocks we included under the stock combo box. Whenever a stock gets chosen, it will showcase all its related stock prices and it will automatically pop-up a chart showcasing the stock's close prices over time.

Web Scraping

The team Web Scraped from *Financial Content* to retrieve current and historical stock prices for the tech stocks selected to research.

Apple Stock was scraped from designated URL:
https://markets.financialcontent.com/stocks/quote/historical?Symbol=537%3A908440

Google Stock was scraped from designated URL:
https://markets.financialcontent.com/stocks/quote/historical?Symbol=537%3A29798540

Amazon Stock was scraped from designated URL:
https://markets.financialcontent.com/stocks/quote/historical?Symbol=537%3A645156.

Intel Stock was scraped from designated URL:
https://markets.financialcontent.com/stocks/quote/historical?Symbol=537%3A941595

Microsoft Stock was scraped from designated URL:
https://markets.financialcontent.com/stocks/quote/historical?Symbol=537%3A951692

Arrays

In order to move over the data we scraped into a database, the team improvised and created an array to populate our stock database. In its entirety, the array sees the text file of the html code from each of the stock's websites. Then it uses regex to pull the attributes we wanted from the website such as the date, opening price, closing price, volume, and days change. After regex finds that information, it implements it into the array that was created. Once the array has been populated, it will move over that information into SQLite.