Web App: Stock Metrics

Team: Error404

Team Members: Jaynish Shah, Monark Modi, Yanjie Gao, Wenbo Xie, Bin Yu

GitHub: https://github.com/monarkmodi/Stock-Metrics

Overview: Our web application is a cloud based stock analysis tool for the users which helps a user manage their stocks and keep a track on their net profit/loss. It will use machine learning techniques to analyze existing metrics and try to predict the future value of the stock. It fetches from a list of 8000 stocks across Nasdaq, NYSE exchanges and performs support vector regression based machine learning on historical values like open, close, high, low and volume to forecast the next day price change. The web application also works as a simple watch list and portfolio application to keep an eye on user's stocks and compare performances. For now, we don't have any changes from our project proposal since the Project 1 submission.

Design Overview: To implement the login logout functionality, we created login form to store the username and password in the database. This function also included session management where only the logged in users could access certain pages like Portfolio or Stock Metrics. We created a group called StockHolders which included all the users for the website. Once the user logged in, the user was able to view its data such as the portfolio value, search for stocks and a list of all the searched stocks.

Problems/Successes: To get all the stock data we first started working with Yahoo finance API but later on we found out that the API was shut down. So, we had to find another open source API. As a result, we implemented Google finance. While working with the google finance API, other team members were not able to run the server. To fix that issue, we had to run the setup.py file and then run the server. While working with the forms we were able to save the stocks in the database but we not able to save it in the portfolio page.

Team Choice: For the final submission, our team wishes to implement the complex stock metric indicators such as RSI (Relative Strength Index), OBV (On Balance Volume), SMA (Simple Moving Average) and Aroon Indicator. For the implementation, we will have to learn about those formulas and implement them in our project.

Individual Write-ups:

Jaynish Shah: Until now in the project, for project1 I worked on couple of the UI designs implementing basic CSS and worked on the write-up part on all the three projects. For second project I discussed different data types and models that we could implement for the project along with the data model overview. I populated the admin page with mock data. I created a dynamic content page showing basic information such as the number of stocks in the database etc. For project 3 I worked on the write-up and researched a little bit about using the finance API and tried solving few bugs. Percent work: **15%**

Monark Modi: For the third project, I started off by implementing session management and login required function to limit the access to certain pages. Then I started working on the part 2 of the project and developed the complete portfolio page that uses Google's finance api to retrieve the stock data and showcase it on portfolio page. The issue I faced was that I was able to save the info to django database but I was not able to show on the html page. Also, I rewrote the API script in client.py to get personalized data that we needed. Then, I wrote the complete stock metric page that calculates stock indicators based on different formulas. Lastly, I worked on github merge issues due to different codes written by other teammates. Percent work: **30%**

Yanjie Gao: For project 1, I finished some pages of the UI design with CSS and HTML. For project 2, initially, I finished the Part 0 which configures the environment and starts the Django project. Then, I designed the data model diagram and implemented most of data models using Django's Object Relational Mapping with Python. Next, I performed the database migrations and registered these data models for the sake of making them accessible in the Django admin site. Finally, I created a superuser account with username and password, and finished a few part of the write-up. Meanwhile, I also discussed with the team about the implementation of templates and views in part 2, and figured out how to display the field values in the view. For project 3, firstly, I created some users and a group in the database. Then I enabled the authentication and implemented the login/logout functionality with Django form. Next, I applied the static assets to render the login/logout user interface. In the end, I finished the password reset function and rendered the relevant UI. Percent Work: **20%**

Bin Yu: For project 1, I finished index html with css file. For project 2,I participated the group discussion about model design and make some models. For project 3, I accomplished some authentication part like log in and log out pages myself, but they are not used in the team. I work with Wenbo to accomplish the order function that is one of our interaction with forms. I add the code in forms.py for our order feature where we want the users to be able to buy stock and show the prices. Percent work: **15%**

Wenbo Xie: For the project 1, I was responsable to make the registration page and its css file based on the the user interface designed by the other group members. For the project 2, I worded on designing the models for this application with the other group members in one of the group meeting. Also, I implemented the function of counting the number of visitors for our webpage, but unfortunately, it ended up being discarded. For the project 3, Bin Yu and I worded on the part 2 and added ‘Order’ page to our website. I created the template of this page and implemented Django form. Fundamentally, the symbol and number of shares will be collected and passed to Django server via the form generated by Django Form API and return the amount the user needs to pay for the shares of the stock the user want to buy and the basic stock information. To implement this, I defined function called ‘order’ in views.py and used OrderForm implemented by Bin Yu in forms.py to created text-fields and ‘Calculate’ button of form and the google finance api provided by Shah and Modi to find the stock information. In the end, I am still working on ‘Order’ button. After clicking it, user can see the the stock it just bought in Stock Page. Percent work: **20%**