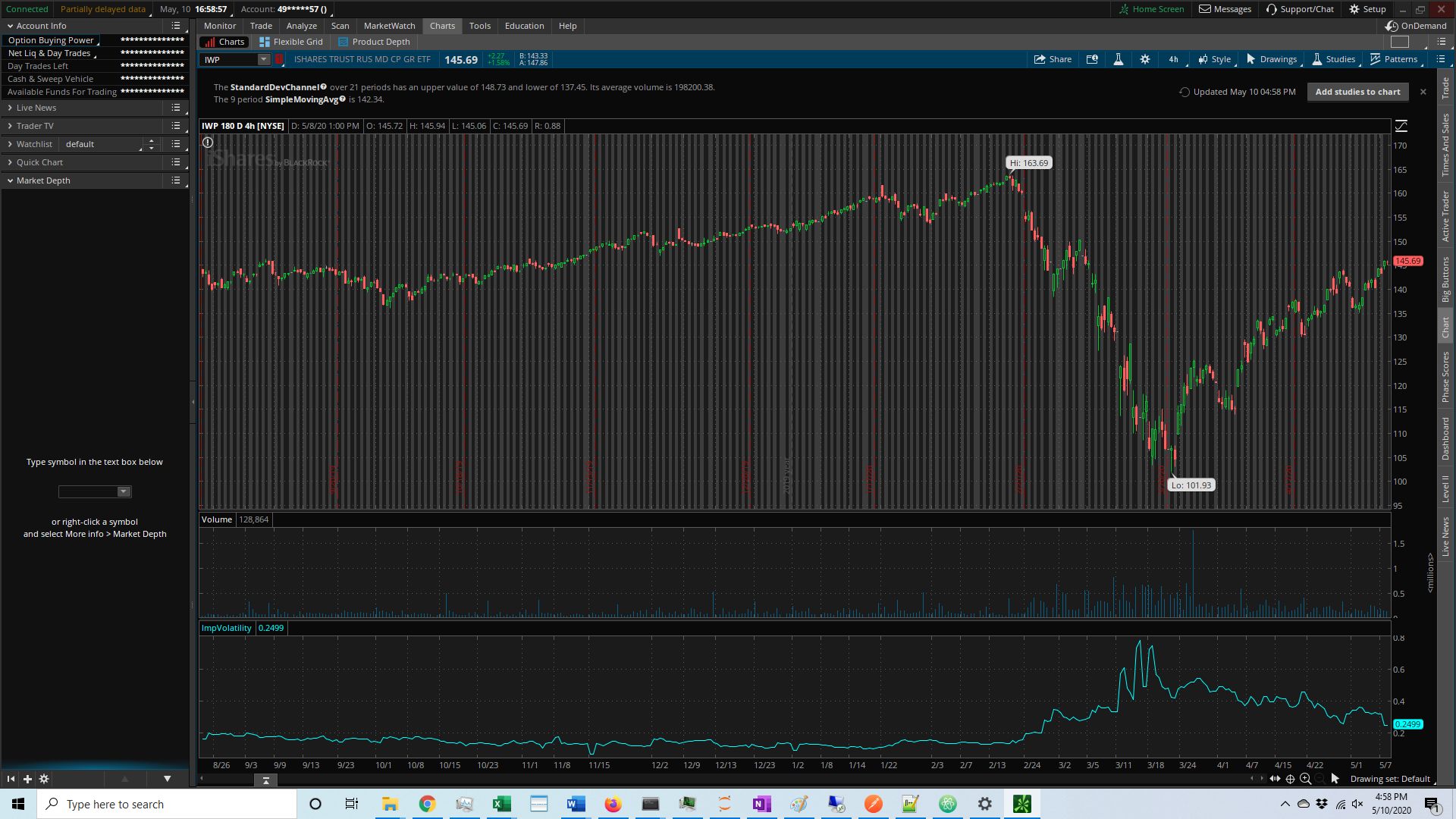
**Introduction – High level description of the webapp**

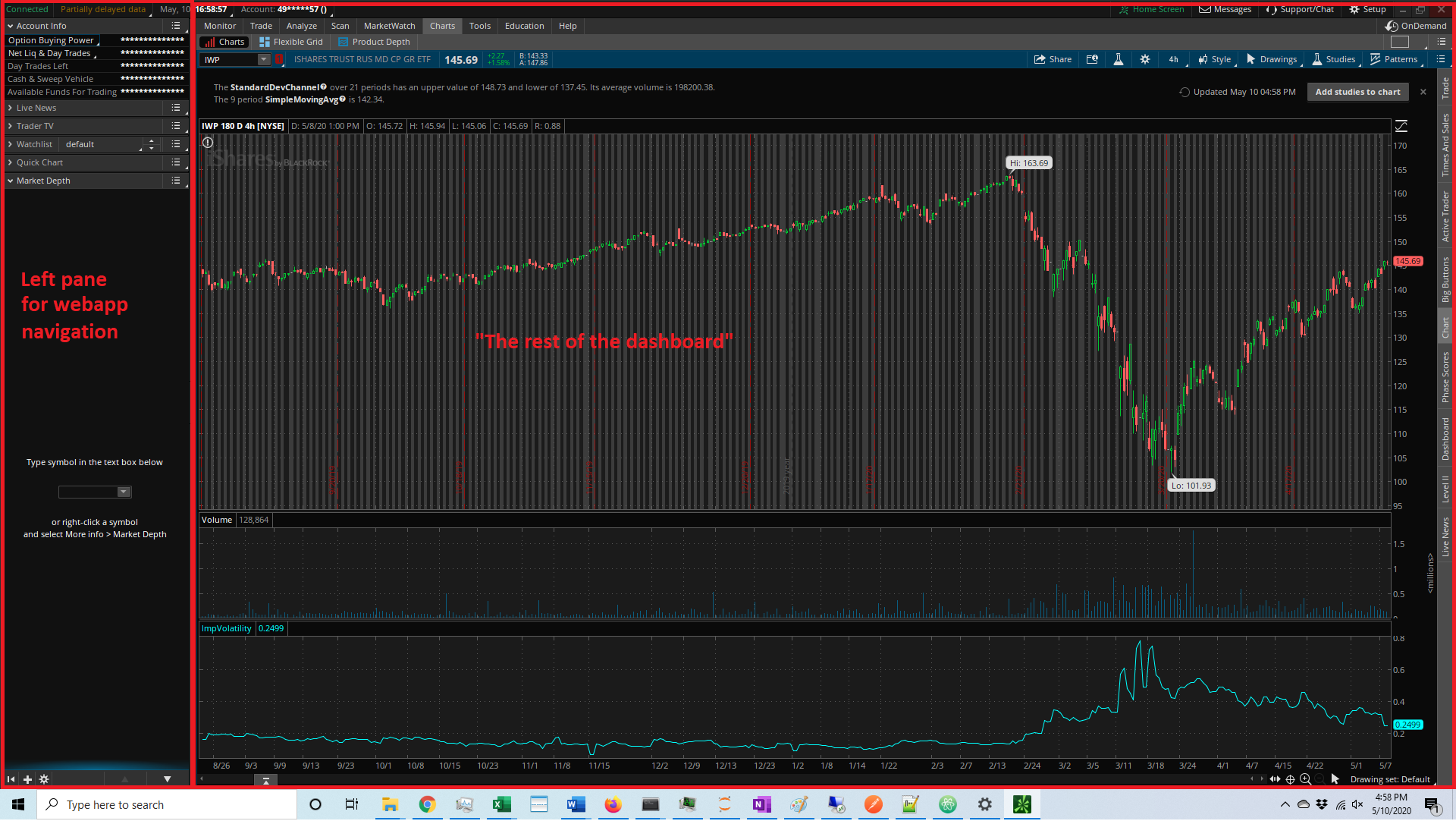
For the main webapp, I would like to use a framework that’s available off the shelf. I’m thinking either Dash or Bokeh. I saw that you had featured both on your site. I’m very much leaning towards Dash because it’s made by Plotly and it just seems like an easier and quicker solution to get to a good/clean webapp up and running. Can you let me know what you think?

Design

For design, I would like to largely copy the theme that another popular stock analysis software uses. I’ve included a screenshot below:



There isn’t much to necessarily copy here… but overall, I would like to leverage the general dark/black color scheme and the frame proportions. Where there’s a panel on the left that takes up ~15% of the total width and then the rest of the information takes up the remaining 85%. I’ve outlined in red what I mean in the context of the first screenshot.



You can see in the above screenshot how I’ve divided up the two sections. I consider the left useful for general navigation between the sections and the right useful for displaying the relevant information for whatever section of the webapp is open.

\*\*Borrowing an existing design that’s already been made…

The following dashboard might be able to provide a template that we can leverage to move quicker:

<https://dash-gallery.plotly.host/dash-daq-satellite-dashboard/>

This existing site isn’t completely how I’d like the webapp to look… but it’s the closest one I could find to my vision out of all the ones listed on Plotly’s gallery: <https://dash-gallery.plotly.host/Portal/>

I think this example is a good template to copy because it provides us a left pane which isn’t too large and also the site loads nicely on both my computer and smartphone.

The source code for it is available here: <https://github.com/plotly/dash-sample-apps/tree/master/apps/dash-daq-satellite-dashboard>

Other webapp features:

1 – Data source

I am working to finish up a relatively simple API that the webapp will be able to use to get all data needed for the graphs. I’ve been following along using this instruction video and its been a lot of fun!.. <https://www.udemy.com/course/rest-api-flask-and-python/>

With that being said… I should have the api finish this week. It isn’t too complex overall. The main reason I opted for this is that I’m thinking ahead and trying to follow best practices for software development: <https://www.jinfonet.com/resources/bi-defined/3-tier-architecture-complete-overview/>

My thinking is this design will scale better long term.

The example Jupyter notebooks I sent over use CSV files that I think are fine for the short term. I’m trying to get started on the graphs first and this way we’re not waiting for the API to be finished to be able to begin development. I assume it wont be difficult to switch the data source of the graphs once the API is ready.

2 – User functionality

Id like to have traditional user functionality for the app. By this I mean… sign up / login / forgot password, etc.

I’d also like to have a payment module so that users can be charged a monthly fee for access to the application. I’m thinking Stripe will be best to manage the actual billing, but I’m still researching how this functionality can be incorporated. (<http://stripe.com>)

One advanced feature that I’d like to have is the ability for a user to have referral codes where they can invite other users to sign up. I’d like to grant each new user two referrals. My actual plan is to have the webapp closed off to individual signups initially… unless someone has received a referral code from an existing user. This way it’s a somewhat private application. I can then decide what the monthly fee should be at a later date.

I’m still looking into how to accomplish this and am certainly open to any suggestions.

We can focus on the user functionality later on in the project but I wanted to make you aware of it in case its relevant for design and planning purposes.

3 – Hosting

I think pythonanywhere would be a great place to host the app. What do you think? Thank you for introducing me to this service… they seem awesome!

\*\* To dos\*\*

I think the best order of operations is to…

1. Design / build the functionality of each of the two graphs
2. Integrate the API into the graphs
3. Design / build the DASH webapp (and layout)
4. Integrate payment and user functionality

What do you think? Please let me know your feedback when you can! I appreciate it!!