For my project, I decided to take the difficult challenge of creating a bot to automate stock trading. I wanted to do this so I could learn more about programming software applicable in the real world, and learn more about how the stock market functions. I originally wanted to write this in Golang, but I reluctantly settled on Python because there's so many really good libraries for crunching numbers, and the API I used for actual trading has a nice official Python wrapper. I used a brokerage called Alpaca, which is built for trading with API calls, so that was perfect for my purposes. I use a service called Polygon, which is used by websites like Robinhood, to get all of the data. It goes through a list of tickers I've laid out in the config file and for each of them, gets the barset of the last many hours, and then I run it through a function called `bbsqueeze()` which looks for a squeeze with bollinger bands. Bollinger bands are a common indicator to determine whether a certain stock is underbought or overbought. If a stock is overbought, it will likely correct itself (price goes down), and if it's underbought it will likely correct upward (price goes up). It can also be used to detect strong movements, which is how I'm using it for the squeeze check. If there is a large enough difference in the upper and lower bands, my function notices and takes note. If afterwards the chart consolidates, meaning the price moves very little, it is likely going to break upwards or downwards. This can be determined by which bollinger band it breaks. When my algorithm detects a consolidation, it buys shares and waits for a break. This algorithm is good at detecting strong movements and reacting. One of the more valuable skills I've learned through this project, is utilizing a Python library called Pandas, which is a library used for working with lots of numbers, especially in a table format, which stocks have a lot of. This is easily the largest scale project I've ever worked on and I started to run into issues with my lack of planning early on, and the code kept getting messier and messier, and more hacky. Admittedly, if I were to continue to work on this, I would probably start by rewriting most of it. Nonetheless, I feel like I've learned a lot from my experience writing this and learning more about the stock market and how to write modular code.