Project Summary

Project overview	The goal of this project is to build a quantitative research tool that allows users to slice and dice time series data and generate trading ideas. The tool should have a user-friendly UI similar to TradingView, where users can select date ranges and frequency of candles. Additionally, users should be able to add time series signals on top of candles and make queries to filter data based on specific criteria.
Requirements	a. Develop front-end UI similar to TradingView , where one can select a date range and the frequency of the candles • frequency can be set to 1min, 15 min, 1h, 1 day, 1 month b. The ability to add time series signals on top of the candles • moving averages of the close price of the last K candles, where K is to be specified by the user c. Ability to make queries like the following: • Find all time series (from multiple assets and multiple time ranges) where close price was 20% higher than the 20 SMA (simple moving average), where 20 refers to the number of candles used for the moving average • Find all time series where volume was 3x above the 7-periods volume SMA AND SIGNAL_X is below 5%
Scope	Data Storage: The first component of this project is data storage. The data will be stored using Amazon RDS for PostgreSQL, which provides scalable and durable object storage. The data for each asset including time series, prices, volumes, and some commonly used SMA will be stored there for easy access. Back-end Data processing: For the backend, there are mainly two data processing jobs: updating the graph and answering queries. We can use an ETL (extract, transform, load) pipeline to achieve this. • For graph update, such as switching granularity of the graph, or adding signals, we first extract data from our database, transform it by adding the data into our current graph, and load it by sending it to the front end. • For queries, we're first extracting the client's filtering requirements, transforming it into a corresponding SQL commands, and finally load the data to front end.

	This can be done using AWS Glue, which is a fully managed ETL (extract, transform, load) service that makes it easy to move data between data stores. However, this can also be done manually. Front-end UI Development: For the front-end, there are two main components for our dashboard. First, we need to construct a candle stick graph which allows user interaction, including zooming in/out, dragging, and viewing data when hovering. It also needs to have a section where user can set parameters for the graph, such as adding SMA signals, changing frequency of the candlesticks, choosing asset and so on. Second, we need to a query section where the user can have filter their requirements. The design should be intuitive and easy to understand. It also needs to capture many types of query. We can use Plotly Dash for the front-end development, a Python framework for building analytical web applications. It provides a set of high-level components for creating interactive data visualization and a user-friendly interface for creating and customizing web applications. Security: The final component of this project is security. We'll require the users to login to use the application. We can integrate the application with the same secure authentication method that's used in Monoceros.	
Project Timeline	• Set up PostgreSQL database on Amazon RDS • Set up version control system (Git) • Develop code that can populate real-time data automatically into the database • Have an initial design of front end UI	1 week
	• Get approved on UI design • Develop front-end code for building the entire UI	1 week
	• Develop back-end code for candle stick graph by building a ETL pipeline • The candle stick graph should support features including adding signals, changing candle frequencies, and basic interactions on the graph • Optimize graph loading time	1 week
	• Develop back-end code for the query system by building a ETL pipeline • It should support a wide range of queries	1 week

	• Create testing cases • Make pull request and have a code review • Deploy with EC2 using AWS cloud services • Set up security authentication	2 weeks