**1. Objectives and Aims.**

**1.1 Problem identification**

In the current stock trading world, manual application of trading strategies can be very exhausting and time consuming to professional traders. To clarify, day traders spend endless days and nights perfecting their trading plans, only to get on their computer at market open and stare at a candlestick chart for hours waiting for the perfect opportunity to arise and apply their strategy. Moreover, traders find themselves mentally exhausted as they usually spend two to three hours staring at a screen to perform trades which last a few minutes, sometimes seconds. In addition, another issue with the current trading proposal is the difference in time zones. Thus, users outside the United States of America are often unable to trade due to the stock market opening late at night which makes it difficult and unhealthy for them being awake staring at charts at 2am.

**1.2 Aims**

The primary aim of my application is to automate the aforementioned, by setting up bots to execute trading orders when the opportunity occurs, through a user-friendly experience. Further aims of the project are; implementation of real money trading through APIs such as the one Binance provides, adoption of the software by hedge fund companies and automate their employees’ tasks.

**1.3 Objectives**

The application will be called Strategify and will be developed using the quasar framework. The main objectives are the following.

* Create a user-friendly front end using quasar’s utilities and Vue js.
* Quasar uses node js to its core which will be used to create the back end which connects the app with the database
* Google’s firebase will be used to store all the application’s data
* Binance API will be implemented to fetch live candlestick data

**1.4 Brief description of Strategify**

I aim to develop a web application platform where users can automate their own strategies for cryptocurrency trading and let a bot trade for them while they are away or sleeping. The application will also provide already existing strategies made by professionals for the users to copy. The platform will also provide fictional U.S. dollars for the users to practise and test their strategies with. Moreover, each user can have active and idle strategies which they can create through the application’s UI. Active strategies will place market orders when the requirements specified are met. Whenever a user creates a strategy, they have to place BUY and SELL requirements based on indicators and price point. They also have to choose a ticker pair from the ones provided by the application (e.g., BTCUSDT as in bitcoin against the US dollar stable coin Tether) and also the amount of money they want to trade. An example strategy the user could create is for a BUY order of 100$ to be placed when the RSI indicator hits the value 30 and a SELL order when it hits 70 on ETHUSDT at the four hour candlestick chart.

**2. Literature Review**

**2.1 Introduction**

Researching on the development of such a system can be very challenging because it requires the intercommunication of many different technologies. Moreover, most of the technical research was taken on mastering the Quasar framework along with Vue js and also understanding websockets and data fetching from external APIs. Furthermore, a lot of research was undertaken on understanding the fundamentals of investing and trading such as technical analysis and grasp understanding of complex candlestick charts components.

**2.2 Quasar Framework**

Quasar is a javascript framework which aims to provide developers with the ease of having a cross platform app for web mobile and desktop application, all in one codebase [2]. In addition, Quasar uses Vue js for front end development and also provides several components on their website for easy UI creation. The reason for me choosing this framework is the responsive element it provides, its ease of understanding and use but also secures the scalability potential of the application as it can easily be further developed in other platforms other than web, with just a few weeks in the codebase. The Quasar Framework was first released its 1.0 version in 2019 which makes it a newly released framework [1]. Thus, knowledge on its implementation was not taught to me in school or university. With decent understanding of basic vanilla javascript you could get around the framework, but to further expand my knowledge on it, I have undertaken an online course which goes over all the fundamentals of the framework and explains well on how to scale to a larger-scale application [3].

**2.3 Fundamentals of Trading.**

The world of finance and decentralised finance specifically has caught my interest in the past few years which made investing in assets and also researching start-ups a hobby for me. Although, I have a good understanding of different blockchain technologies, I also need to get a good understanding on trading specifically in order to understand my target audience and correctly develop such a platform. For obtaining such knowledge, I have undertaken another online course which has its primary focus on day trading basics [4].

**2.4 APIs and web sockets.**

The first time I had ever used APIs was at university which made me comfortable using them also helped me understand their functionality. Moreover, to make sure I completely understand the APIs needed for the app I had created a mini trading bot with hardcoded strategies in order to make sure I grasp the knowledge needed to create the application’s back end. The bot was created by following a tutorial video [5]. Although I have successfully created a bot using the desired APIs, it was made using python. In addition, I aim to learn how to implement the same APIs with javascript for my project’s needs.

[1] <https://dreamonkey.com/en/blog/quasar-conf-2020-all-the-latest-news/>

[2] <https://quasar.dev/introduction-to-quasar>

[3] <https://www.udemy.com/course/quasarframework/>

[4] <https://www.youtube.com/playlist?list=PLic3jeMsmNn3bHRQw_RaSaTBF5ryOj3lQ>

[5] <https://www.youtube.com/watch?v=GdlFhF6gjKo&t=3361s>