



Final Report

Web Application that uses algorithms to automatically identify candlestick patterns on financial charts.

CM3203 – One Semester Individual Project – 40 Credits

Author: Shakur Duale

Supervisor: Federico Liberatore

Moderator: Parisa Eslambolchilar

Abstract

Candlestick patterns are the language of financial charts. Created by Munehisa Honma in the 17th century. They are the staple tools used by traders in financial analysis and price predictions. With how important and widely used they are, it is surprising that there are not many applications that support their identification. Additionally, even though they are so important many traders do not use them or have not heard of them. A lot of financial traders invest based on the news they hear about a company or currency. This is not a bad reason to invest but it should be supported with extensive financial analysis which is not possible to do without knowledge of candlestick patterns. So why are there not many applications that teach them? This report describes, assesses, and evaluates the “Candlestick Finder” application. The primary feature of this application is to identify candlestick patterns from a chart automatically. The secondary feature is to explain the common candlestick patterns at a level where beginner traders can understand and learn.

This study will evaluate the usability of the application. Agile methods were used in the planning and development of the application. Test cases and analysis of requirements were used to evaluate the results. User interface design heuristics are considered and evaluated throughout the report. The aims, objectives and requirements were chosen after being researched thoroughly. The result of the study shows that the application has various shortcomings in implementation and testing, however, the final product achieves the requirement and the usability of a professional product in the future is worth considering.

Acknowledgements

There are so many people that have helped me this semester and in the entire course as a whole but there are a few special mentions. I would like to give huge thanks to my supervisor, Dr Federico Liberatore, for guiding me throughout this project. There were many times when I felt anxious or unsure of my progress and the meetings always helped me to remain calm and keep my direction. I would also like to thank Dr Nervo Verdezoto Dias and Dr Parisa Eslambolchilar for their help during the Design Thinking module in the Autumn semester. That module really helped me to flesh out and focus my idea for the final project.

To all my family and friends, I thank you so much for all the support you have given me and for the encouragement to push through and keep going in tough times. Thank you to my mum for always believing in me. Thank you to my dad for being a role model to look up to.

Table of Contents

INTRODUCTION.....	5
PREFACE.....	5
PROJECT SCOPE AND PROBLEM.....	7
AIMS AND OBJECTIVES	7
AUDIENCE	8
ASSUMPTIONS.....	8
IMPORTANT OUTCOMES.....	9
BACKGROUND.....	9

PREFACE.....	9
STAKEHOLDERS	9
FINANCIAL MARKETS	9
<i>What are financial markets?</i>	10
<i>What is supply and demand?</i>	10
<i>What affects supply and demand?</i>	11
<i>How do financial traders profit?</i>	12
CANDLESTICK PATTERNS	12
<i>What's a candlestick pattern?</i>	13
<i>The Psychology behind Candlestick Patterns</i>	14
EXISTING WORKS	15
<i>Candlestick Pattern Finder by (niko_83, 2020)</i>	15
Advantages (+)	15
Disadvantages (-)	16
Evaluation	16
<i>(CandleScanner, n.d.)</i>	16
Advantages (+)	17
Disadvantages (-)	17
Evaluation	17
<i>Proposed Solution</i>	17
Constraints	18
RESEARCH QUESTIONS	18
APPROACH	18
SOFTWARE DEVELOPMENT METHODOLOGY	18
<i>Agile Development Methodology</i>	18
<i>Why Agile?</i>	19
Advantages (+)	19
Disadvantages (-)	19
<i>Applying Agile Methods</i>	19
TECHNOLOGIES AND TOOLS USED	20
<i>Design and Prototyping</i>	20
SPECIFICATION AND DESIGN	20
PREFACE.....	20
FUNCTIONAL REQUIREMENTS	20
NON-FUNCTIONAL REQUIREMENTS	21
USE CASE DIAGRAMS.....	22
<i>First Use Case</i>	22
<i>Second Use Case</i>	23
<i>Third Use Case</i>	24
SITEMAP	24
FLOWCHART.....	25
PROTOTYPE.....	26
<i>Homepage</i>	27
<i>Chart Page</i>	28
<i>Explanation Page</i>	29
USER INTERFACE DESIGN (HEURISTICS)	29
<i>Visibility of System Status</i>	29
<i>Match between system and the real world</i>	30
<i>User control and freedom</i>	30
<i>Consistency and standards</i>	30
<i>Error Prevention</i>	30
<i>Recognition rather than recall</i>	31
<i>Flexibility and efficiency of use</i>	31
<i>Aesthetic and minimalist design</i>	31
<i>Help users recognize, diagnose, and recover from errors</i>	32
<i>Help and documentation</i>	32

IMPLEMENTATION	32
PREFACE.....	32
SETUP.....	32
Python	32
Atom	33
Flask Setup.....	33
Bootstrap Template.....	34
Website Design.....	37
Version Control.....	37
PROJECT LAYOUT	37
Python files	38
Templates and CSS	38
Extra/Other Files.....	38
FIRST ITERATION	38
Design.....	39
Twelvedata	39
Plotly.....	39
Development	39
Setting up Twelvedata	39
Setting up Plotly.....	42
Testing.....	43
Output	43
Iteration summary.....	43
SECOND ITERATION	43
Design.....	44
Annotations	44
Development	44
Testing.....	45
Iteration Summary.....	46
Problems faced	46
THIRD ITERATION	47
Design.....	47
Development	47
Testing.....	48
Exception Flow.....	49
Iteration Summary.....	49
Problems faced	49
FOURTH ITERATION	49
Design.....	50
Research	50
Development	50
Creating the explanation pages	50
Adding the navigation.....	51
Testing.....	51
Iteration Summary.....	51
FINAL PRODUCT REVIEW	52
Home Page	52
Chart Page.....	53
Explanation Pages	54
RESULTS AND EVALUATION.....	54
TEST CASES.....	54
First Test Case.....	54
Second Test Case	55
Third Test Case	56
Fourth Test Case.....	58
Fifth Test Case	59
Sixth Test Case.....	60

Seventh Use Case.....	61
Test Case Evaluation.....	62
Advantages (+).....	62
Disadvantages (-)	63
Test Case Summary	63
REQUIREMENTS ANALYSIS.....	63
First functional requirement.....	63
Second Functional Requirement.....	64
Third Functional Requirement.....	65
Fourth Functional Requirement.....	67
Requirements Evaluation	69
HEURISTIC EVALUATION	69
Visibility of System Status.....	69
Match between system and the real world.....	70
User control and freedom.....	70
Consistency and standards.....	71
Error Prevention	71
Recognition rather than recall.....	72
Flexibility and efficiency of use.....	72
Aesthetic and minimalist design.....	72
Help users recognize, diagnose, and recover from errors	73
Help and documentation.....	74
RESULTS AND EVALUATION SUMMARY	74
FUTURE WORK.....	74
ADD MORE CANDLESTICK PATTERNS.....	74
MORE EXAMPLES IN EXPLANATION PAGES	74
IMPROVED PERFORMANCE.....	74
JAVASCRIPT	75
SUPPORT AND RESISTANCE CALCULATIONS.....	75
CONCLUSIONS.....	76
REFLECTION ON LEARNING.....	76
DIFFICULTY OF IDENTIFYING CANDLESTICKS	77
TIME MANAGEMENT	77
SUPERVISOR MEETINGS.....	77
AGILE METHODOLOGY	77
DIFFICULTIES IN MOTIVATION	78
WORKS CITED.....	78

Introduction

Preface

Bartering, trading, and markets have existed in society for hundreds of millennia. Researchers found evidence of early Homo Sapiens with items that were not local suggesting that they must have traded with other groups (Polianskaya, 2018). As time progressed, humans used products as a staple of value such as rice (e.g., this product is worth 5kg of rice). Eventually, humans started using hard currency and then paper currency. Methods of trade have consistently evolved until today. In recent years, the biggest evolution in trading is the introduction of technology. Transactions are made electronically, and the use of physical currencies decreases by the day even digital currencies are being

created called Cryptocurrency (Kaspersky, n.d.). With trading evolving every day, in turn, traders need to improve their trading tools as well. Most financial traders with experience trading stocks, forex, cryptocurrencies etc, will use candlestick pattern analysis. This is the most popular tool used in financial analysis but with technology transforming so many aspects of trading, why do we need to do this manually? If there was an application that analysed financial charts automatically it would be another step forward in the evolution of trading.

Today there are 13.9 million online financial traders in the world (Singh, n.d.). This includes stock trading, forex trading and cryptocurrency trading. In 2020, global trade fell by 8.9%, the steepest drop since the global financial crisis (Bank Of England, 2021). The start of the pandemic in 2020 also had a huge impact on the cryptocurrency trading sector but in a different way. The cryptocurrency market saw a huge influx of new traders. When the pandemic erupted, Bitcoin – the world's first cryptocurrency – could be purchased for about \$7,300. Today, the very same token costs more than \$46,800 – a staggering 640 percent rise (Jabotinsky & Sarel, 2021). This huge increase in the price of Bitcoin goes to show just how much the market has boomed. A lot of people who have previously never traded before had started with Bitcoin, Ethereum, Dogecoin or another cryptocurrency during the pandemic. A CNBC survey found that 1 in 10 people currently invest in cryptocurrencies (Reinicke, 2021). With so many people getting into trading, how do we teach them how to read financial charts and understand candlestick analysis? This project details the development of an application that provides a solution to this question.

Candlestick Patterns originated in Japan in the 17th Century (Raut, 2020). At the time rice trading dominated the market and had become more important than actual currency. Munehisa Honma was considered to be the most successful financial trader of all time and also the first person to use candlesticks in financial trading. Although he was familiar with supply and demand tactics in trading, he primarily sought to understand the psychology and emotions behind decision making in financial trades. Through his research, he developed his own trading system known as candlestick analysis. This trading system was applicable in all financial markets and revolutionised financial trading. Munehisa Honma had amassed in his life a huge sum equivalent to 10 billion US dollars. He was later given samurai status for his achievements. For a long time, Japan was using candlestick analysis and only in the 1980s did it reach the western world. Today, candlestick analysis is a standard tool that financial analysts use to formulate their price predictions.

How can we improve candlestick analysis? The project I am developing does not aim to replace or evolve candlestick analysis but support it. I will be creating an application that processes the data of a financial chart and uses algorithms to identify candlestick patterns. The application will then display the financial chart with indicators showing where the candlestick patterns are. There will also be pages explaining what each of the candlestick patterns mean for the user to reference. This project aims to support the user in formulating accurate and reliable price predictions that miss little to no indicators as well as helping the user to learn and remember all the candlestick patterns.

Project Scope and Problem

Most financial traders today use candlestick pattern analysis as their template for any price predictions or market analysis. However, there are many kinds of candlestick patterns and if you are not actively looking for one, they can be easy to miss. To support financial traders in their market analysis I will be creating an application that is essentially a 'Candlestick Finder'. It will process the data of the user's chosen financial chart and identify and display the candlestick patterns on the chart for the user to see. This will protect traders from missing potential price indicators which will, in turn, solidify and improve their price predictions. It will also stop traders from having to analyse the chart again and again which is monotonous and time-consuming.

When someone is new to financial trading everything seems a bit overwhelming. There are so many different markets such as Forex, the Stock Exchange or cryptocurrency, and there are also a lot of different trading types such as leverage trading, futures contracts etc. Candlestick patterns are amazing in that they can be used in any financial market or trading type. This is because they are the language of a financial chart. Therefore, for beginner traders, it is essential that they first learn the language of financial charts. The application that I am developing does not only identify candlestick patterns but also explains what these candlestick patterns are. It explains what they mean, and it explains how to spot them. This feature is important in providing new traders with a solid foundation and preventing them from trading in the dark.

Aims and Objectives

The primary aim of this project is to create an application that analyses the data of a financial chart and compares the high, low, open, and close prices of each trading session to calculate and identify the candlestick patterns on the chart and visualise the data for the user.

- 1.) The first aim is to acquire the data of a financial chart through an API.
 - a. The first objective is to make a connection between the server and the API.
 - b. The second objective is to create a search engine that allows the user to choose any currency pair such as 'BTC/USD' which is the cryptocurrency Bitcoin to the US Dollar or GBP/USD which is British Sterling to the US Dollar. The user should be able to choose currency pairs from multiple markets.
 - c. The third objective is to collect the data and visualise it for the user in a suitable format.
- 2.) The second aim of the project is to show the user detailed definitions and explanations of the candlestick patterns identified:
 - a. The first objective is to research all the candlestick patterns and make detailed notes.
 - b. The second objective is to create a page for each of the candlestick patterns detailing what they are, how to spot them and what they mean with some examples.

- c. The third objective is to create links so that when a financial chart is analysed and displays the candlestick patterns, there are links to the definition pages.
- 3.) The third aim of this project is to create a website that visualises the data in an easy to use and attractive design:
- a. Design a simple and neat financial chart to display the candlestick patterns that look familiar. Should look like the standard financial charts on most popular trading sites.
 - b. The pages containing the explanations of the candlestick patterns should include a lot of figures, examples and sketches to help with explanations. (Not filled with text)
 - c. An attractive home page that explains what the website is with as few words as possible.
- 4.) The fourth aim of this project is to identify the candlestick patterns from the financial data provided.
- a. Process the data into a suitable format e.g., a CSV file or a pandas dataframe.
 - b. Produce algorithms that accurately identify candlestick patterns from the data provided.
 - c. Display the identified candlestick patterns on a financial chart for the user to see.

Audience

This project is targeted toward two different audiences. The first audience that this application will target is veteran traders. The reason they would want to use the application is to reinforce, verify and support their financial analysis and provide another supplementary tool for their trading. This is through the primary functionality of identifying candlestick patterns. The second audience that this application will target is amateur or beginner traders. The reason they would want to use this application is to learn about candlestick patterns and how to use them. This is through the secondary functionality of explaining the candlestick patterns. However, veteran traders may find use in the secondary functionality as a way to recap the candlestick patterns. Likewise, beginner traders would begin to use the primary functionality more as they improve and begin formulating their own financial analysis and price predictions.

Assumptions

There are some things that I understand about my project being the one creating it. However, from an outsider's perspective, some parts of the project will not make sense due to a lack of contextual knowledge. In this section, I list some assumptions for the user to make sense of things.

- This project is a prototype or a proof of concept. I would not have time to show every single candlestick. I would also not be able to design perfect algorithms to identify the candlesticks. However, this prototype will have these features at the level of a prototype, not a complete product. An additional feature that a complete product should have is the ability to calculate support and resistance levels.

- When I mention beginner traders, this could mean people that have been trading for multiple years. These traders might have an extensive economic background and understand supply and demand. However, to be able to read and understand a financial chart, the trader must understand candlesticks and candlestick patterns. Without this, their financial analysis and price predictions are much weaker.

Important Outcomes

For this project, there are three very important outcomes.

- 1.) The user can search for the chart of their chosen currency pair.
- 2.) The user can identify candlestick patterns on the chart.
- 3.) The user can see explanations and examples of the candlestick patterns.

Background

Preface

Before starting the “Candlestick Finder” application, it is important to research the primary topics relating to the application such as financial markets and how they work and function. This is important to know as it helps to understand not just how candlestick patterns will affect the market but also why. Another topic that requires in-depth research is candlestick patterns themselves. This research will include the history of candlestick patterns, explanations of a few candlestick patterns, how they work and the psychology behind them. Finally, it is imperative to find existing works that tackle a similar, if not the same, problem. From evaluating these works I aim to take their strengths and avoid shortcomings that they encountered.

Stakeholders

When considering stakeholders, what I mean is the different parties that are affected by the problem. Thus, the primary stakeholders, in this case, would be the target audience. To reiterate the problem once again: Candlestick patterns are one of the fundamental tools professional financial traders use to formulate their financial analysis and price predictions. That is because it is the language of financial charts. Just like you learn English to read an English book, you learn candlesticks and candlestick patterns to read financial charts. Therefore, why are there so many beginners and amateur traders that are unaware of this? Additionally, with the huge development of technology in the last few years why are there no electronic tools to support traders in identifying candlestick patterns. This is the problem. As you can see, beginner and professional traders are both stakeholders as they are both impacted by this problem.

Financial Markets

Financial markets refer broadly to any marketplace where the trading of ‘securities’ occurs, including the stock market, bond market, forex market, and derivatives market, among others. Financial markets are vital to the smooth operation of capitalist economies (Hayes, 2022). This definition essentially explains what a financial market is but to fully understand it we must understand what the term ‘securities’ means. The term "security" refers to a

fungible, negotiable financial instrument that holds some type of monetary value. It represents an ownership position in a publicly-traded corporation via stock; a creditor relationship with a governmental body or a corporation represented by owning that entity's bond; or rights to ownership as represented by an option (Kenton, 2021). The two primary types of securities are equity and debt. Debt refers to money/assets owed and equity refers to money/assets owned. The financial markets that we focus on for the scope of this project are stock exchange markets, forex markets (currency trading) and cryptocurrency markets.

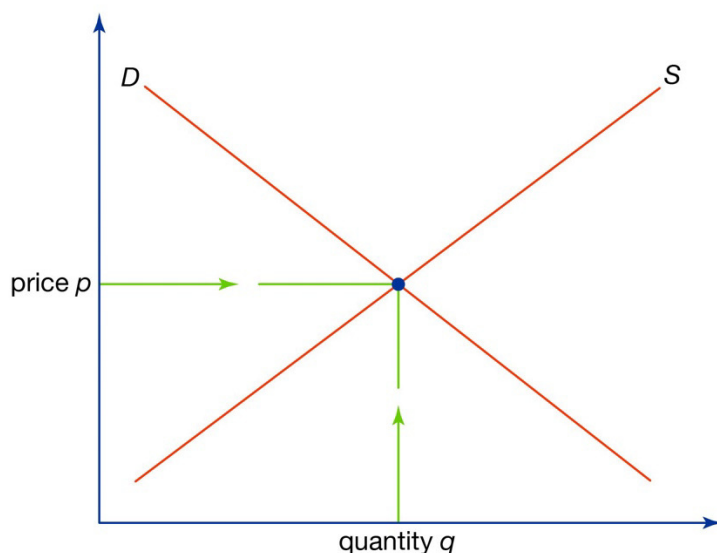
What are financial markets?

Financial markets facilitate the trade of financial assets but how does this allow financial traders to profit? This is because of supply and demand. If demand for a product is high and supply is low then the product will have a higher value e.g., gold. On the other hand, if the supply for a product is plentiful and the demand is lower than the supply then the product will have a lower value e.g., plastic bags. Of course, this is a simple explanation that does not account for numerous factors, however, all other factors simply influence supply and demand whereas supply and demand is the foundation that any market is built upon.

What is supply and demand?

The law of demand says that at higher prices, buyers will demand less of an economic good. The law of supply says that at higher prices, sellers will supply more of an economic good. These two laws interact to determine the actual market prices and volume of goods that are traded on a market (Fernando, 2021). The two laws interacting are best shown in Figure 1 below.

Supply and demand



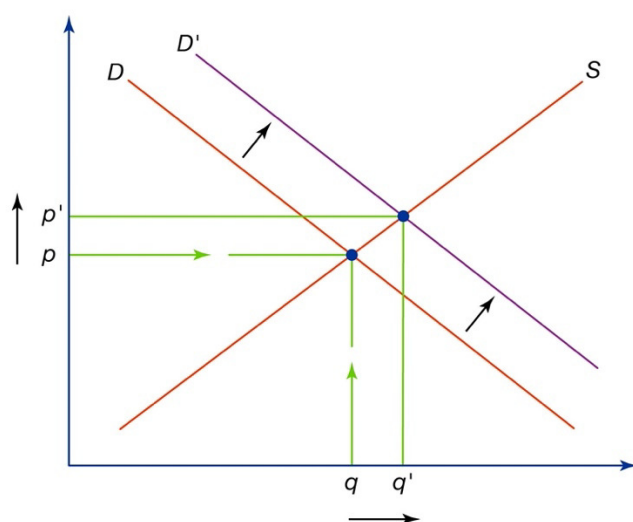
© 2013 Encyclopædia Britannica, Inc.

Figure 1: Supply and Demand Chart

This chart shows the price p of the product on the vertical y -axis and the quantity of the product q on the horizontal x -axis. The line labelled S is the supply, and it shows that as the price increases the quantity supplied increases as stated in the law of supply. The line

labelled D is the demand, and it shows that as the price increases the quantity supplied decreases. Where demand and supply for a product meet is the equilibrium and this decides the price and quantity supplied. This is how supply and demand affect financial markets. When more people start buying shares of a company the price of each share will increase. This is because demand is exceeding supply. The price will stop increasing when the market reaches equilibrium again. This is best explained in Figure 2 below.

A shift in demand



© 2013 Encyclopædia Britannica, Inc.

Figure 2: Shift in Demand Chart

This chart shows what an increase in demand looks like. As the demand has increased, the equilibrium between supply and demand has increased as well. This in turn caused an increase in price and an increase in quantity supplied. A recent example that everyone has experienced is the increase in the price of masks and the quantity produced as well because of the pandemic. This is a clear example of external factors influencing markets but there are many other factors to consider as well.

What affects supply and demand?

Supply can be influenced by many different factors. An increase in production costs can cause a decrease in supply. A recent example of this is rising gas prices. A lot of product prices have increased to compensate for this, but this will in turn cause demand to decrease. If fewer people are buying and they make fewer profits, they will eventually be unable to sustain the amount they produce and decrease it. Other factors include taxes and regulations, technological improvements, changes in the workforce available etc. Demand also has many influencing factors. Seasons affect demand. Umbrellas are more popular in rainy seasons. Time affects demand. Train tickets cost more at peak times as there are much more people taking the train at those times. Substitutes affect demand. A monopolised product will earn more than a product with competitors. Complimentary products affect demand. An increase in iPhones sold would increase the amount of iPhone chargers sold. However, all these factors may apply to normal markets, but financial markets are slightly different.

How do financial traders profit?

A financial market is where people meet to buy and sell financial assets such as currencies. So, when a lot of people start buying a currency that is an increase in demand for that currency and the price of it will increase. This has been the case with cryptocurrencies such as Bitcoin which has had a huge increase in price in recent years (Jabotinsky & Sarel, 2021). On the other hand, if lots of people start selling Bitcoin then this is a decrease in demand and the price will begin to drop. Financial traders aim to make a profit by buying when the price is low and selling when the price is high. So, they make it their business to understand the market and pay attention to many factors that affect the demand for the currency or stock. I will be producing an application that supports financial traders in their price predictions through candlestick pattern analysis.

Candlestick Patterns

Candlestick patterns created by Munehisa Honma have become a staple of financial markets and financial analysis (Raut, 2020). It also allowed financial traders to predict price changes and market trends caused by external factors from a chart. Figure 3 below shows what a candlestick chart looks like.



Figure 3: Example of a Candlestick Chart (The Data Visualisation Catalogue, n.d.)

The candlestick chart is a type of time-series chart that displays the prices over time. Each candlestick shows 4 different prices as explained in Figure 4 below.

What is a candlestick?

Japanese candlesticks are formed using the open, high, low and close of the chosen time frame.

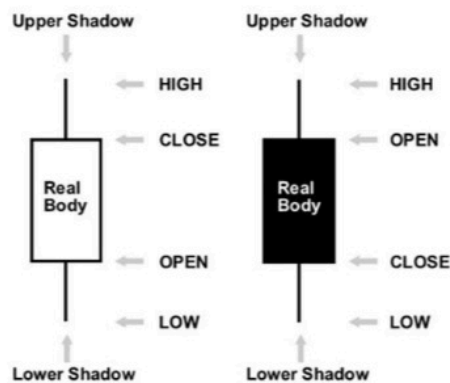


Figure 4: A Candlestick in Detail

Figure 4 shows how a candlestick works. Each candlestick shows the open price which was the price at the start of the trading session and the close price is the price at the end. The high is the highest price reached in that session and vice versa. A candlestick is green, white or what we call 'bullish' if the closing price is higher than the open price. A candlestick is red, black or what we call 'bearish' if the closing price is lower than the open price.

What's a candlestick pattern?

A candlestick pattern is a pattern on the chart that signifies a market change. These candlestick patterns have been researched and reviewed many times since they were created by Munehisa Honma. An example of a candlestick pattern is the Doji candlestick pattern shown in Figure 5 below.



Figure 5: Doji Candlestick Pattern (Raut, 2020)

The Doji candlestick pattern can be identified because the candlestick pattern occurs when the opening price and closing price are equal or close to equal. This gives the image of a candlestick with no body. When this candlestick pattern occurs at the top of an uptrend or the bottom of a downtrend it signifies a trend reversal. This is shown by the example in Figure 6 below.

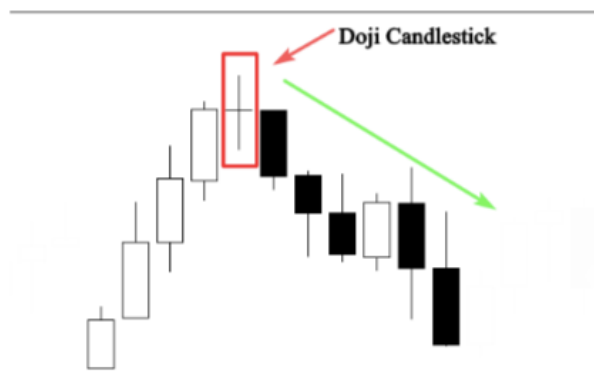


Figure 6: Doji Bearish Reversal (Raut, 2020)

The Psychology behind Candlestick Patterns

The example above shows that at the top of the uptrend buyers pushed the price to the high, but it got pulled down back to the open by the sellers. The sellers then pushed the price down to the low, but it got pulled back up to the open by the buyers. Candlestick Patterns were created by Munehisa Honma as he was trying to understand the psychology of financial traders and the market. The Doji candlestick pattern represented indecision. Where buying and selling pressure is equal and the market is uncertain or resting after an intense period of trading. This can mean a reversal as shown in the graph above or a weakening of the previous trend, but it primarily represents indecision (Pines, 2022). The second example of a candlestick pattern is shown in Figure 7 below.

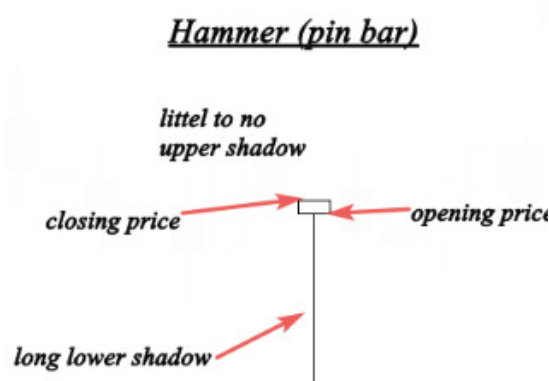


Figure 7: Hammer (Pin Bar) Candlestick Pattern (Raut, 2020)

The Hammer or Pin-Bar Candlestick pattern has a very small body with little or no upper shadow. Its defining characteristic is the very long lower shadow. It is a bullish candlestick pattern that represents sellers losing interest. The sellers bought out all their assets causing the price to drop steeply to the low. This is a bullish signal that buyers can look forward to opportunities with sellers having left the market. Every candlestick pattern has psychology behind it. They have specific shapes and characteristics for a reason. This is what makes them so powerful. Financial traders can understand what is going on in the market and what traders are thinking through the financial chart! The solution I offer with my application is an automatic analysis of a financial chart to discover the candlestick patterns. Without missing a single candlestick pattern on the chart, a financial trader can understand the market on a deeper level.

Existing Works

Although it was an idea that I came up with, it is not the first of its kind. There have already been applications produced that offer similar functionality and support to my application. I will be going through a couple of these applications and explaining what the goal of the application is, evaluating the advantages and disadvantages and then comparing it to the solution that I will be offering.

Candlestick Pattern Finder by (niko_83, 2020)



Figure 8: Candlestick Pattern Finder by - (niko_83, 2020)

Figure 8 above shows an application created by niko_83 that has the same aim as my solution. The aim is to identify and display candlestick patterns to the user from the chart. He has completed this through some algorithms and his application is built into TradingView's website. TradingView already has some in-built algorithms that allow the user to identify candlestick patterns, however, there are only a select few and there is a limited amount that you can search for before you need to buy a premium account. niko_83 has used TradingView's algorithms to create an application that identifies every candlestick pattern on the chart instead of TradingView's version where you select them one at a time. On the chart, he has identified the candlestick patterns with little captions containing the name of the candlestick pattern identified in abbreviation.

Advantages (+)

- By having the application built into TradingView's website, he can use TradingView's charts which makes the application look very nice.
- The captions are clearly visible, and colour coded.
- The captions are not too big but not too small, a nice fitting size.
- The application allows the user to only indicate bullish patterns or only indicate bearish patterns or both.
- The algorithms used to identify the candlestick patterns are from TradingView and have been verified many times, so they are reliable and accurate.

Disadvantages (-)

- The captions show the abbreviation of the candlestick pattern. It is not clear which candlestick pattern is identified and there is no explanation of what the abbreviation means.
- The application does not allow the user to pick and choose which candlestick patterns are displayed. Although this was not the goal of the user during implementation, it is something an application like this should have.
- The captions are quite far from the candlestick patterns and because of this, it is a bit confusing which candlestick the caption is referring to.
- There is no feature to support beginner traders in learning about candlestick patterns.

Evaluation

The application shows all the candlestick patterns quite clearly on a TradingView's chart which looks very good. However, there are not many options for the user and although the display of the captions is nice, it is confusing as to what they mean and which candlestick they are referring to. In the solution that I offer the user will be able to pick and choose which candlestick patterns they would like to display; the captions will contain the full name of the candlestick pattern and will be close to the candlestick, so it is clear which one is being referred to.

(CandleScanner, n.d.)

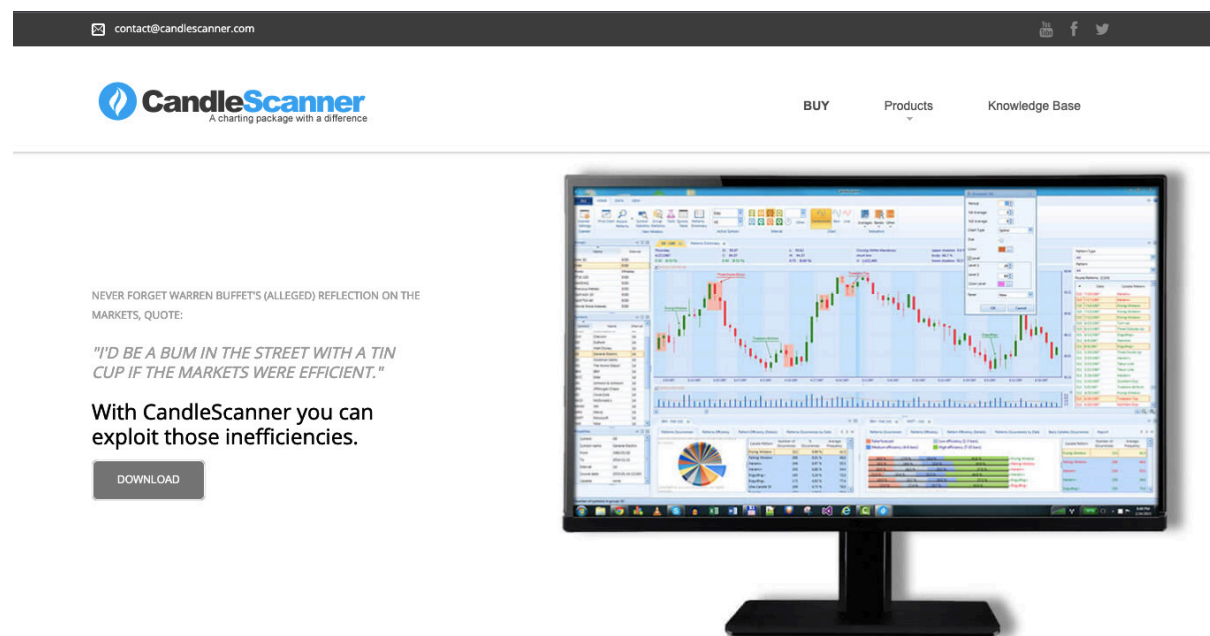


Figure 9: CandleScanner Application

Figure 9 above shows a professional product called CandleScanner by a company of the same name. This product gives the user a desktop application with a detailed user interface that allows the user to analyse financial charts in detail. The aim of this application is the

same as my own. It aims to provide the user with an application primarily for identifying candlesticks.

Advantages (+)

- “There are over 100 built-in candlestick patterns at your fingertip” (CandleScanner, n.d.).
- The user can find a detailed explanation for each of the candlestick patterns. The application also aims to teach candlestick patterns
- “You have access to historical and current data from over 70 exchanges worldwide and many asset classes: stocks, indices, bonds, currencies, cryptocurrencies, or commodities” (CandleScanner, n.d.)

Disadvantages (-)

- This is a desktop application which means it requires downloading before use which is time-consuming. Additionally, it can only be accessed from the computer it's downloaded on.
- The application is professional and created for a profit. The user will have to buy the application which is a barrier to many people
- According to the website, the technical requirements are as follows: “You need a computer running on Windows 7/8/10 (64-bit) + Internet connection” (CandleScanner, n.d.). This means that users on Mac, Linux or other operating systems cannot use this application.

Evaluation

This application encompasses everything that I would like to include in my solution such as the candlestick pattern recognition, the access to multiple exchanges and the explanations for candlestick patterns. It has a professional design and in-depth functionalities. However, this application is limited by its impracticality. It is a desktop application that can only be used with certain operating systems and costs money. If you are away from your computer, you cannot use it. The solution I offer is a website. By nature, you can access it anywhere for no cost and it requires no download. This means that the audience that I can reach has the potential to be far higher than the application shown above.

Proposed Solution

The solution that I propose identifies candlestick patterns from a financial chart using my custom algorithms. These algorithms will analyse the data of each candlestick and that candlestick will be highlighted on the chart in some way if it fits the criteria of a candlestick pattern. The user should be able to pick and choose which candlestick patterns are displayed. The user should be able to go to definition pages for each candlestick pattern that explains what they are, how to spot them and what they mean with some examples, images and sketches. The user should also be able to search for the financial chart they are looking for e.g., BTC/USDT or GBP/USD. This solution was initially based on pattern recognition. I used to think that using machine learning to predict future prices would be very useful but as I researched the topic area more and more, I realised that my solution is more beneficial. Financial markets are influenced by so many different factors as explained already, a machine will not predict prices accurately. Additionally, this is not solving a

problem but just attempting to replace humans with machines. The solution I have provided after extensive research is what I believe to be more helpful and practical to the target audience in solving the problem.

Constraints

The primary constraint for this solution is time. Due to the limited timeframe available to complete this project, it will likely have a few design features that could be improved by the end, or the functionalities may not be on the level of some professional applications. For example, for the scope of this project, I will not be using every single candlestick pattern but only a select few more common ones. Another constraint comes from me not having completed a similar level project before. Therefore, there will be a lot of discovery and revised plans meaning that I cannot accurately produce milestones and deadlines.

Research Questions

In order to demonstrate the achievement of the stated aim, appropriate APIs for obtaining relevant and up to date financial data from cryptocurrency, forex or stock exchanges will be researched and the optimal ones will be implemented. This project will identify relevant performance, efficiency, and reliability metrics to enhance custom algorithms. These metrics will also be used to judge whether the algorithms are indeed an improvement and evaluate the quality of the product as a whole.

Approach

Software Development Methodology

For a project of a certain level, if it contains multiple functionalities, files, and software, it is important to have a structure in development. This structure should include planning, design, implementation, testing, evaluation, and maintenance (this step is not considered for the scope of this project). Therefore, it is important to decide on a development methodology to follow. There are many different development methodologies such as the waterfall method, the agile method, the spiral method, the prototyping method etc. For this project, I will be using an agile development methodology.

Agile Development Methodology

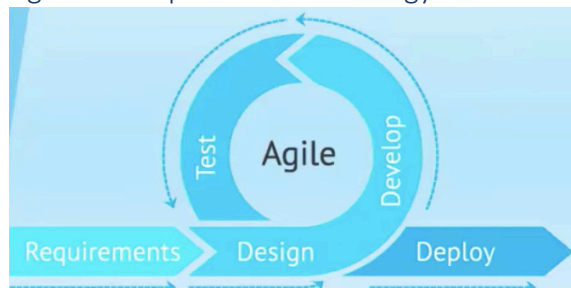


Figure 10: Agile Development Methodology - (Smart Learning Studio, 2020)

The agile development methodology repeats three stages in iterations. This is the distinctive feature of the agile method. This originates from the objective of producing functional code for the customer as possible. Figure 10 above shows the agile development process. Firstly, the functional and non-functional requirements are produced. After that is the iterative

process of design, development, and testing. After all the iterations are completed, the project should only need refactoring, debugging, and refining before a finished product is completed. The final stage is deployment and maintenance, but they are not being considered for the scope of this project.

Why Agile?

There are many reasons for picking this development methodology over others such as the waterfall, spiral and prototype methodology. The primary reasons are shown in the advantages and disadvantages section below.

Advantages (+)

- More flexible
- More likely to find errors
- Efficient
- Testing is quick

Disadvantages (-)

- Can be tiring repeating the same steps
- Meeting deadlines becomes difficult

The advantages above show that the agile methodology is very efficient. The disadvantage of monotonous development is minor and the struggle to meet deadlines should not be a problem as I have already made an initial plan that has been verified. I did not pick the waterfall development methodology because it is not flexible. One of the four main values of the agile methodology is 'Responding to change over following a plan'. I have never created a project of this size before and will likely be using a variety of new software as well, therefore, it is important for me to have room to change the design as I go. I did not pick the spiral methodology as it is quite complex and can cause too much time to be spent on documentation. Another one of the four main values is "Working software over comprehensive documentation". I did not pick the prototyping methodology because I do not work well off prototypes and struggle to make them. I have used them for this project but that is just to get a clear overview of what the product might look like. It would be too time-consuming to create accurate prototypes of what I want the final product to look like.

Applying Agile Methods

For the most part, I will be following the agile development methodology described in Figure 10. However, I will be creating my own specification and design which encompasses the requirements section and parts of the design process such as sitemaps, flowcharts, prototypes and technologies and tools used. After this is the Implementation section where I will be doing the iterations. Each iteration will work towards completing a use case. For the testing, I will be using test cases. This is a session-based testing method. This project is going to be the biggest I've done with software I have never used before. Therefore, test-driven development is not suitable. During the iteration, I will only do simple session-based tests with screenshots showing if the use case is complete. The detailed test cases will be completed in the Results and Evaluation section after Implementation. These test cases will check that the functional and non-functional requirements acceptance criteria have been met.

Technologies and Tools Used

Design and Prototyping

For my design and prototyping, I will be using Axure RP9 and Figma. Axure RP9 is a desktop application for prototyping that I have used for multiple modules already. It has many advantages such as the templates, and interactions that can mimic functionalities. This tool is most useful for the early prototypes to figure out what the functionalities might look like and the navigation between pages. Figma is the second prototyping tool and is more useful in the later stage of prototyping. That is because it allows you to directly grab the HTML and CSS for the designs you make. However, Figma is only a supplementary tool that I may use if I am not satisfied with the Bootstrap templates I find online.

Specification and Design

Preface

In this section, I will be going through the design of the application. This design will include diagrams, requirements, use cases, user stories, heuristics, prototypes and more. However, as I have chosen the agile development methodology, I will be designing the software to be implemented in detail in the implementation section, not this one.

Functional Requirements

FR1: The application must allow the user to choose which chart they want to analyse.

The user must be able to choose the chart they want to analyse for this application to have any use.

Acceptance Criteria:

- The application has a search bar that displays the chart of the currency pair put in the search bar. The currency pair will be abbreviated e.g., "GBP/USD".
- The time between the user pressing search and the corresponding chart being displayed must be within 5 seconds.
- The application must allow the user to pick currency pairs from at least one financial exchange market e.g., forex, cryptocurrency.

FR2: The application must display a candlestick chart to the user

For the user to be able to identify candlestick patterns on the chart of their choice they need to be able to see the chart.

Acceptance Criteria:

- The candlestick chart displayed must be interactive. It is not enough for the chart to be a still image.

- The chart must have clear axes with the price on the left and the time on the bottom.
- The chart may also contain titles and labels.

FR3: The application must allow the user to see candlestick patterns on the chart

This is the primary functionality of the application. The chart must display candlestick patterns clearly for the user to see.

Acceptance Criteria:

- The candlestick patterns must be identified clearly on the chart with accurate labels.
- The candlestick patterns identified must be consistent with what the candlestick pattern looks like.
- The user should be able to choose which candlestick patterns to display and hide the ones they no longer want to see on the chart.

FR4: The application must allow the user to see the definitions and explanations for the candlestick patterns

This is the second primary functionality on the website that helps the user to learn about the candlestick patterns, what they mean and how to find them.

Acceptance Criteria:

- The explanation pages can be accessed from any page.
- Each explanation page must show clearly what the candlestick pattern looks like.
- Each explanation page must explain clearly what the candlestick pattern means.
- Each explanation page should contain examples.
- Each explanation page could contain real-life examples.

Non-Functional Requirements

The following non-functional requirements detail how the system should perform in terms of usability, reliability, efficiency, and maintainability.

Usability:

- The user should be able to access every other page on the website from each page.
- The contents of each page should not overwhelm the user with large blocks of text.
- Every functionality should include some visual direction on how to use them on the website.

Reliability:

- The user should be able to select any currency pair from at least one financial market exchange.

Efficiency:

- The time between the user pressing search and the corresponding chart being displayed must be within 5 seconds.
- The time between the user choosing a candlestick pattern to display on the chart and the candlestick pattern being displayed on the chart should be below 3 seconds.
- The chart data should be accurate and equivalent to the actual currency pair picked.

Security:

- The application should not take any data from the user.

Use Case Diagrams

First Use Case

Use Case ID:1	Use Case Name: Get the chart of the chosen currency pair	Rating: Must
Description of Purpose: The user wants to see the candlestick chart of the chosen currency pair, so this use case describes the steps taken to do so.		
Basic Flow: The most common form of interaction between the actor and the system. <ol style="list-style-type: none"> 1. Step 1: Start on the Homepage. 2. Step 2: Enter the currency pair chosen into the search bar e.g., BNB/USD and press the search button. 3. Step 3: You have been redirected to the candlestick chart displaying the currency pair chosen in step 2. 		
Alternative Flow: Alternative paths through the system AC1: If the user is not on the home page but one of the explanation pages <ol style="list-style-type: none"> 1. Step 1: Start on one of the explanation pages. 2. Step 2: Enter the currency pair chosen into the search bar e.g., BNB/USD. 3. Step 3: Press the search button. 4. Step 4: You have been redirected to the candlestick chart displaying the currency pair chosen in step 2. AC2: If the user is not on the homepage but the chart page. <ol style="list-style-type: none"> 1. Step 1: Start on the chart page. 2. Step 2: Enter the currency pair chosen into the search bar e.g., BNB/USD. 3. Step 3: Press the search button. 4. Step 4: You have been redirected to the candlestick chart displaying the currency pair chosen in step 2. 		
Exceptions: Exception handling by the system EX1: The user has entered a currency pair that does not exist. <ol style="list-style-type: none"> 1. Step 1: Start on the Homepage. 		

2. Step 2: Enter the currency pair chosen into the search bar e.g., BNB/USD.
3. Step 3: Press the search button.
4. Step 4: The Homepage is refreshed with a flash message saying something such as "Incorrect currency pair. Please type in this format: BNB/USD".

EX2: The user has typed the currency pair incorrectly into the search bar

1. Step 1: Start on the Homepage.
2. Step 2: Enter the currency pair chosen into the search bar e.g., BNB/USD.
3. Step 3: Press the search button.
4. Step 4: The Homepage is refreshed with a flash message saying something such as "Incorrect currency pair. Please type in this format: BNB/USD".

Non-functional requirements:

Performance: The search should give some sort of response within 5 seconds of the search button being pressed.

Second Use Case

Use Case ID:2	Use Case Name: Identify Candlestick Patterns on the Chart.	Rating: Must
Description of Purpose: The user must be able to choose a candlestick pattern and the one chosen should have all instances of it occurring displayed on the candlestick chart.		
Basic Flow: The most common form of interaction between the actor and the system. <ol style="list-style-type: none"> 1. Step 1: Start on the Chart page. 2. Step 2: Select the Doji candlestick pattern. 3. Step 3: The page has been refreshed with all instances of the Doji candlestick pattern displayed. 		
Alternative Flow: Alternative paths through the system AC1: If the user wants to select multiple candlestick patterns <ol style="list-style-type: none"> 1. Step 1: Start on the Chart page. 2. Step 2: Select the Doji candlestick pattern. 3. Step 3: The page has been refreshed with all instances of the Doji candlestick pattern displayed. 4. Step 4: Select the Morningstar candlestick pattern. 5. Step 5: All instances of both the Morningstar and Doji candlestick pattern are displayed on the chart. AC2: If the user wants to clear the chart first. <ol style="list-style-type: none"> 1. Step 1: Start on the Chart page. 2. Step 2: Select the Doji candlestick pattern. 3. Step 3: The page has been refreshed with all instances of the Doji candlestick pattern displayed. 4. Step 4: Press the clear chart button. 5. Step 5: The chart has been cleared of all annotations and labels. 6. Step 6: Select the Morningstar candlestick pattern. 7. Step 7: The page has been refreshed with only instances of the Morningstar candlestick pattern displayed. 		

Exceptions: Exception handling by the system**EX1: The user has selected a candlestick pattern with no instances on the chart.**

1. Step 1: Start on the Chart page.
2. Step 2: Select the Doji candlestick pattern.
3. Step 3: The page has been refreshed and a flash message is displayed saying something like "No instances of the Doji candlestick pattern found".

Non-functional requirements:**Performance:** The selected candlestick pattern should be displayed on the chart withing 3 seconds of being selected.**Accuracy:** The candlestick patterns displayed should look like the actual candlestick pattern. E.g., If I select Doji, the candlestick patterns on the chart should look like a Doji.

Third Use Case

Use Case ID:3	Use Case Name: See the explanations for the candlestick patterns	Rating: Must
Description of Purpose: The user wants to see the explanations for the candlestick patterns and this use case details the steps taken to do so.		
Basic Flow: The most common form of interaction between the actor and the system. <ol style="list-style-type: none">1. Step 1: Start on the Homepage.2. Step 2: Pick the Doji candlestick pattern in the section under the search bar.3. Step 3: You have been redirected to the explanation page of the Doji candlestick pattern.		
Alternative Flow: Alternative paths through the system AC1: If the user is on the chart page <ol style="list-style-type: none">1. Step 1: Start on the Chart page2. Step 2: Select the Doji candlestick pattern.3. Step 3: The page has been refreshed with all instances of the Doji candlestick pattern displayed.4. Step 4: Press the explanation button next to the Doji button.5. Step 5: You have been redirected to the explanation page of the Doji candlestick pattern.		
Non-functional requirements: Performance: The user should be taken to the explanation page within 2 seconds of pressing the link.		

Sitemap

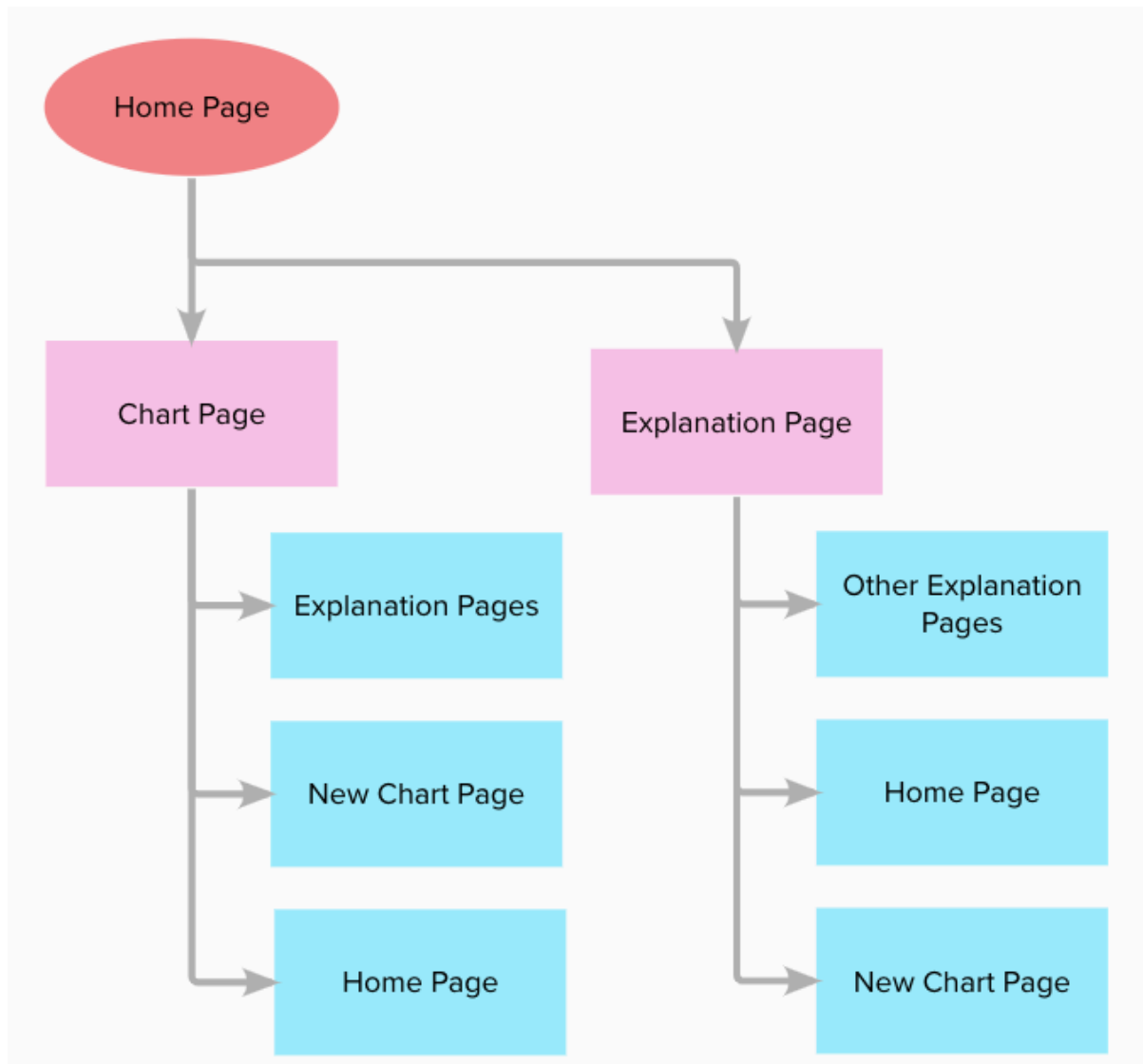


Figure 11: Sitemap of the application

The application I am creating has quite a simple sitemap. From the home chart, you can access the chart page by searching for a currency pair or you can go to any explanation page for the candlestick patterns. From the Chart page, you can choose a candlestick pattern to display on the chart and go to the explanation page of that candlestick. You can also search for a new currency pair and be taken to a new chart page. Finally, you can go back to the homepage. From every explanation page, you can go to any other explanation page, the homepage or search for a currency pair and be taken to a new chart page. Every page on the website has access to every other page on the website.

[Flowchart](#)

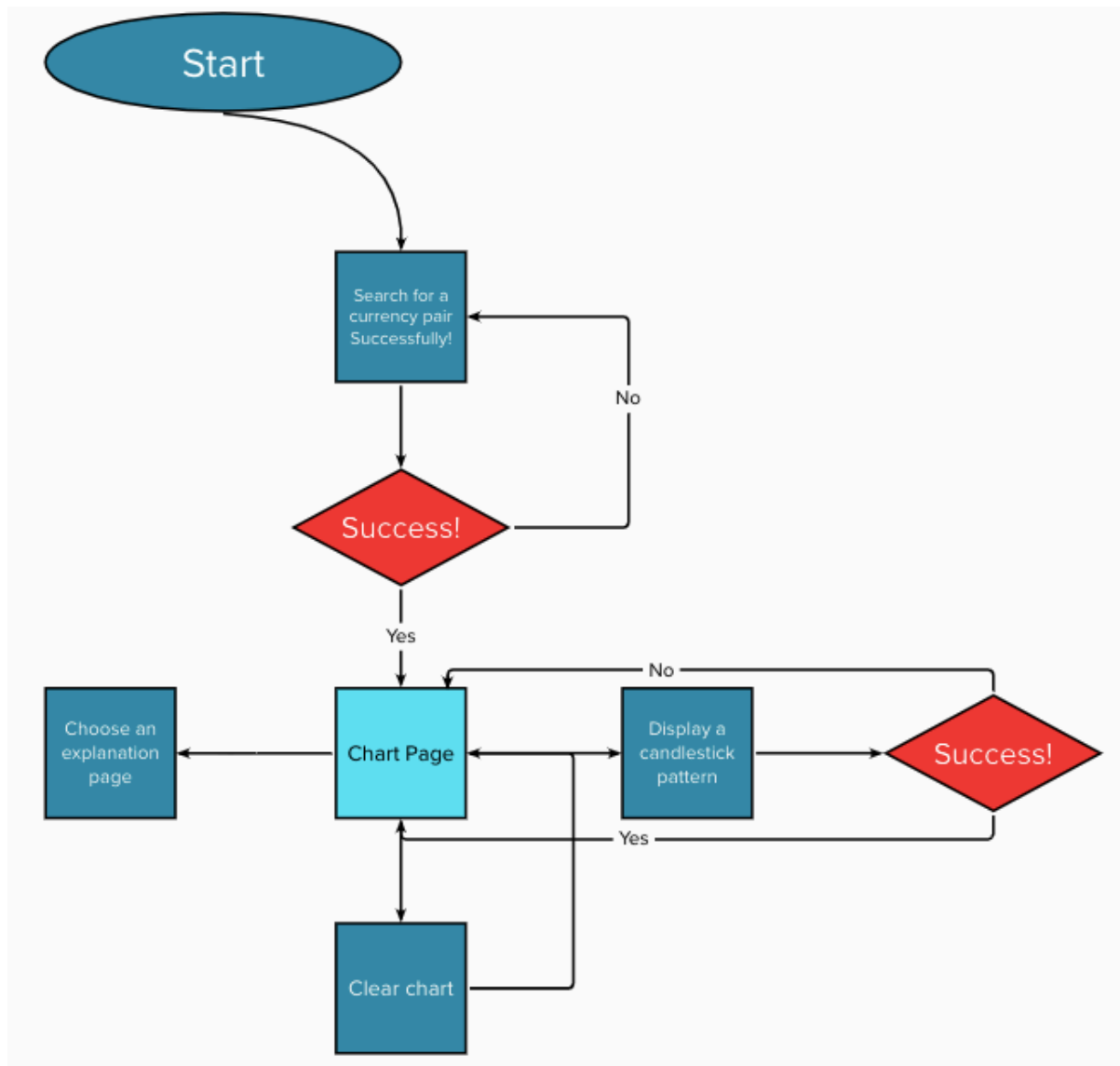


Figure 12: Flowchart

The flowchart in Figure 12 above describes the working process of my application. This application allows you to navigate to any page no matter which page you are on. Therefore, for this flowchart, I did not include navigation, only actual functionalities. Starting on the homepage the first functionality is the search bar. You can search for a currency pair and if you type it in correctly then you will be directed to a chart page displaying that currency pair. If the currency pair is typed in incorrectly or it doesn't exist, you will be redirected to the same page and an error message will be displayed. Once you are on the chart page there are multiple things you can do. You can pick a candlestick pattern to display. If it exists on the chart, it will be displayed. If it doesn't, a flash message will be displayed explaining that it does not exist on the chart. You can also clear the chart of all candlestick patterns displayed as well as go to the explanation pages for the respective candlestick patterns.

Prototype

This section shows my Axure RP10 prototype. The purpose of this prototype is to visualise the functionalities of the application. It is not meant to design the application.

Welcome

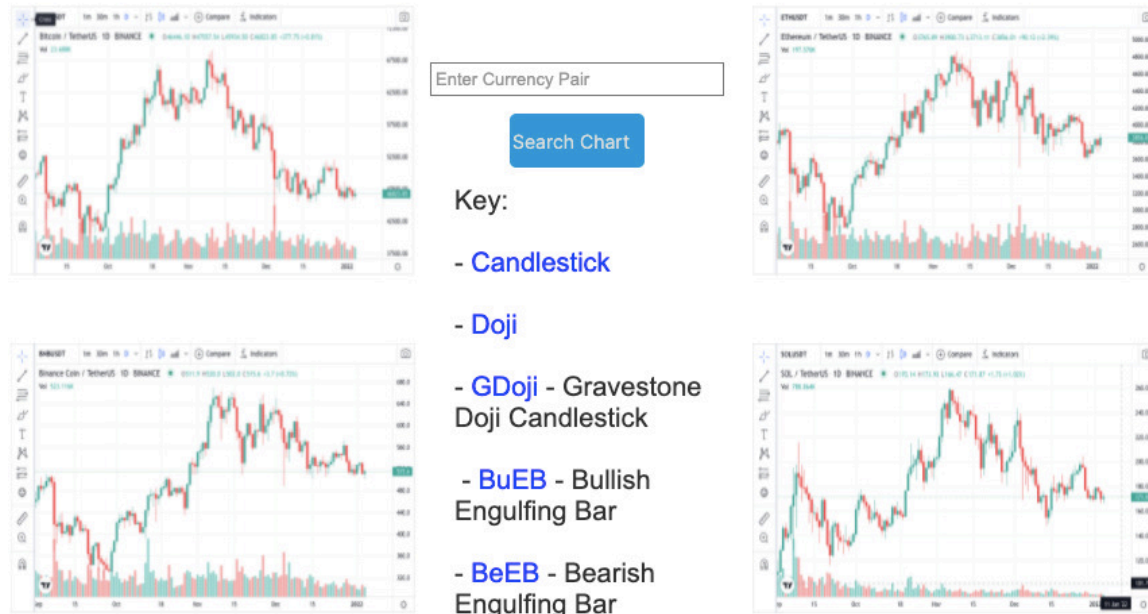


Figure 13: Prototype - Homepage

Figure 13 above shows the homepage of the prototype. The example charts should be clickable and clicking them would in theory take you to the corresponding charts chart page. The user can also use the search bar to search for a specific currency pair. Additionally, there are also links to the explanation pages for some candlestick patterns. This homepage shows all the functionalities the actual will have.

Chart Page

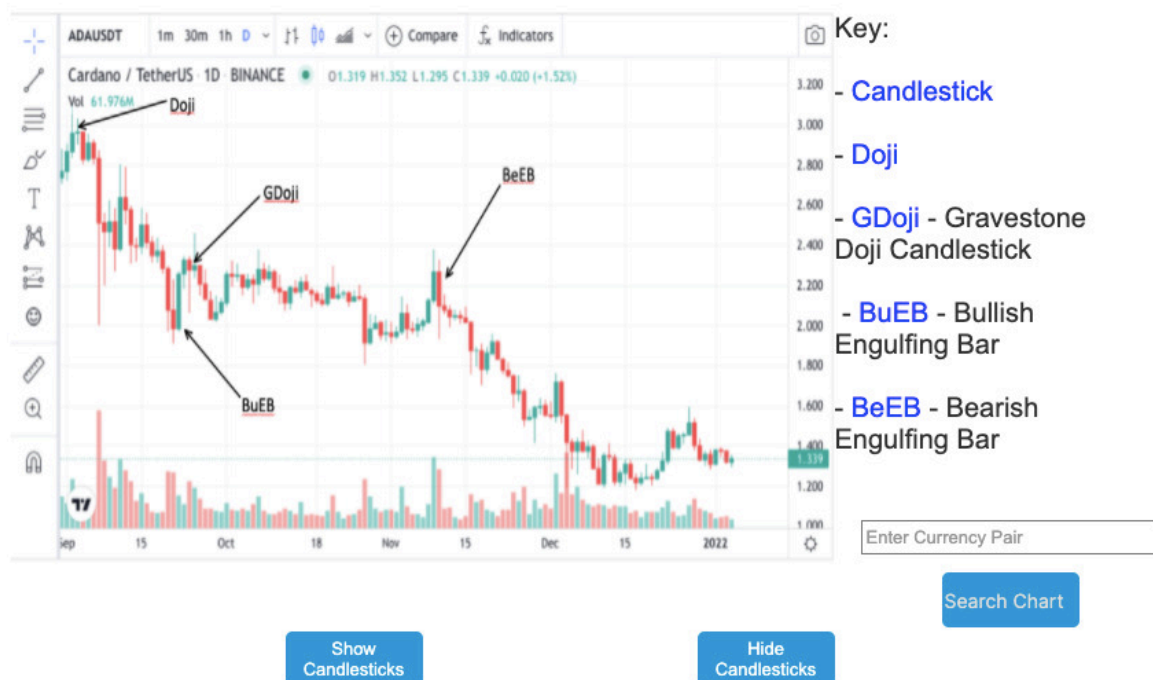


Figure 14: Prototype - Chart Page

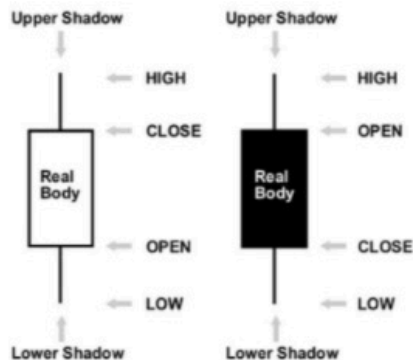
Figure 14 above shows the chart page. This page should display a live and interactive chart. At the bottom, the user can press show candlesticks to display the candlestick patterns on the chart. The 'Hide Candlesticks' button would hide them. In the actual application, the user would be able to pick and choose which candlestick patterns to display. They would also be able to stop showing one candlestick pattern while another is still displayed on the chart. On the right side are the links to the explanation pages and below that is the search bar.



What is a Candlestick?

What is a candlestick?

Japanese candlesticks are formed using the open, high, low and close of the chosen time frame.



A white or green candlestick is called bullish and a black or red candlestick is called bearish.

If the close price is greater than the open price we know that the market was bullish during this time frame and vice versa.

Search Chart

Key:

- [Candlestick](#)
- [Doji](#)
- [GDoji](#) - Gravestone Doji Candlestick
- [BuEB](#) - Bullish Engulfing Bar
- [BeEB](#) - Bearish Engulfing Bar

Figure 15: Prototype - Candlesticks Explanation

Figure 15 above shows the explanation page for candlestick patterns themselves. From this page, the user can go to the other explanation pages as well as search for a different chart. They can also go back to the previous page. This format is the same for the other explanation pages.

User Interface Design (Heuristics)

This section delves into my plans for the design of the application. More importantly than visual design, I will be prioritising interactive design by aiming to follow Jakob Nielsen's 10 general principles for interactive design (Nielsen, 1994).

Visibility of System Status

According to Nielsen's 10 general principles, Visibility of System Status means that the design should always keep users informed about what is going on, through appropriate feedback within a reasonable amount of time (Nielsen, 1994). This is very important because without feedback users will lose trust in the website and believe that it is faulty when no feedback is given. In my project, if a button, link, or something like that is guaranteed to give output such as redirecting to another page, then a simple hover effect changing the colour of the button or link should suffice. However, in the case where a button is pressed and there is a possibility that no output will be given then the user will be given visual feedback in text to explain what happened e.g., a pop-up message.

Match between system and the real world

According to Nielsen's 10 general principles, a match between the system and the real world means that the design should speak the users' language. Use words, phrases, and concepts familiar to the user, rather than internal jargon. Follow real-world conventions, making information appear in a natural and logical order (Nielsen, 1994). This is important as internal jargon isolates your application from a huge audience. This is even more important to me as my application is targeted toward veteran traders as well as beginners. I plan to use real-world conventions and inclusive terminology for the homepage and the chart page. However, for the explanation pages, I believe that it is important to use some trading jargon to allow beginners to both understand concepts and terminology. I want my website to be the bridge between a beginner trader and someone who actively visits trading sites and reads financial articles. For that reason, some jargon is necessary. However, the use of jargon will be limited and hopefully not require the user to have to look up the meaning of the words.

User control and freedom

Users often perform actions by mistake. They need a clearly marked "emergency exit" to leave the unwanted action without having to go through an extended process (Nielsen, 1994). Failure to give the user a 'way out' can cause the user to panic or become frustrated with the application, either one is not desired. In my application, the one thing I ensured, which can be shown by the sitemap, is that the user should be able to access any other page on the website from their current page. I believe this is the highest level of freedom I can provide the user, additionally, the links and navbars look the same on every page. This is shown in the prototype where every page has a search bar and a key that links to the explanation pages of the candlestick patterns. This gives the user a sense of familiarity and therefore, control.

Consistency and standards

It is essential for the application to have consistency and standards. Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform and industry conventions (Nielsen, 1994). A lack of consistency creates a sense of uncertainty and ambiguity for the user. I have already mentioned in the user control and freedom section that I will be using consistency to provide the user with a sense of control through familiarity. That is the same concept here. Familiarity and control go hand in hand. However, consistency within the application and consistency within the industry are two different concepts. My application will utilise navigation bars at the top of the page, search bars and links highlighted in blue at the minimum to give provide the user with familiarity on their first use of the application. This improves the learnability of the application which is critical as I target beginner traders.

Error Prevention

Good error messages are important, but the best designs carefully prevent problems from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action (Nielsen, 1994). In my application, there are no "big decisions" or large forms for the user to complete. I have ensured in the design of the website that the user can navigate freely to any page

which means they can easily go back in the case of a slip (unconscious errors caused by inattention). However, that is not to say that I have not attempted to prevent errors in my design. I plan to have every action the user takes be an informed decision. Thereby, reducing the number of mistakes to a negligible amount. Mistakes are conscious errors based on a mismatch between the user's mental model and the design. The search bar will include hint text so the user knows what they are searching for and the format of the input they should provide. The key will have a header so the users know where the links will take them. There will be a multitude of cases like this where if the action to be taken is not obvious to the user they will be provided with additional text. This ensures that the user's mental model corresponds with the design.

Recognition rather than recall

Minimize the user's memory load by making elements, actions, and options visible. The user should not have to remember information from one part of the interface to another. Information required to use the design (e.g., field labels or menu items) should be visible or easily retrievable when needed (Nielsen, 1994). It should not be a chore for the user to adjust themselves and familiarise themselves with the features of the application. The simplicity of my application's structure makes "recognition rather than recall" a concept that is existent within it by nature. You can see through my prototype that the user can navigate freely to any page. This means that there is no process of remembering how to complete an action. Additionally, there are not many pages in the application. This means that there is no intricate process of navigating pages. Other points mentioned in "error prevention", "consistency and standards" and "match between the system and the real world" show that the user is given ample information and is informed of the decisions they are making before they make them. Therefore, even if they cannot recognise, they can learn.

Flexibility and efficiency of use

Shortcuts — hidden from novice users — may speed up the interaction for the expert user such that the design can cater to both inexperienced and experienced users. Allow users to tailor frequent actions (Nielsen, 1994). Features that improve flexibility and efficiency feel rewarding and satisfying for consistent users of an application and can often cause a user to become a loyal customer. For my application, I am not sure whether this heuristic is applicable. That is because the size of my application is small. Accelerators such as keyboard shortcuts, personalisation and customisation features are relevant for larger and more complete products, but I will likely be unable to reach this stage within the scope of this project.

Aesthetic and minimalist design

Interfaces should not contain information which is irrelevant or rarely needed. Every extra unit of information in an interface competes with the relevant units of information and diminishes their relative visibility (Nielsen, 1994). You can see through my prototype that this is a heuristic that I plan to follow very closely. I do not aim to add features to my website if they do not create a qualitative increase in the value of the product. What I mean by this is that my application has two primary features. One is to identify candlestick patterns on a chart. The other is to explain what those candlestick patterns mean, how to recognise them and how to use them. Any other feature that does not similarly add significant value to the product and the target audience is redundant and does not need to

be implemented. However, I will be including supporting design features that improve the quality of the two existing primary features. For example, I plan to add the candlesticks to be displayed below the chart on the chart page as a card with the image of it shown clearly. This means that when a user wants to see the “Doji” candlestick patterns on the chart they can go to the card below with an image of a Doji candlestick and press show. This is a simple design feature that helps the user to decide the candlestick pattern they want to identify based on an image as well as text.

Help users recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (no error codes), precisely indicate the problem, and constructively suggest a solution (Nielsen, 1994). Error messages are critical when simple interactive feedback is no longer sufficient. In my application, that is rarely the case. Therefore, there will be a few times where the user will receive error messages. The two times that they will receive a message are shown in the use cases. The first case is when the user makes a typo or searches for a currency pair that does not exist. The second case is where the user chooses a candlestick pattern to identify but there are no instances of that candlestick pattern on the chart. Both cases will have text that clearly explains what went wrong, but a solution will not be provided as it is rather obvious how to fix it.

Help and documentation

It's best if the system doesn't need any additional explanation. However, it may be necessary to provide documentation to help users understand how to complete their tasks (Nielsen, 1994). I believe that my application is a system that does not require documentation. This is because the size of the project is small. Therefore, this heuristic is not relevant to the scope of this project.

Implementation

Preface

This section will detail the implementation process of the Candlestick Finder application. This includes the installation and research of the dependencies, the project layout, the implementation of the core functionalities, the use of version control systems and the difficulties encountered that made me diverge from the original design to move forward.

Setup

Python

I decided to use Python as the programming language. There were other choices such as JavaScript. JavaScript would have been very useful in creating an interactive and usable design. JavaScript's ability to update a page live without needing to refresh the page cannot be replaced by any other programming language. Another programming language I could have used was Java as I have the most experience in this programming language. However, Python is the language I have used the most for actual projects and I have also used it for a web application before in the first year. I am familiar with the Python web framework Flask which saves me a lot of learning time. It would take far longer to get a website running with

JavaScript or Java as I have not done it before. I also discovered many useful Python libraries related to the project as well.

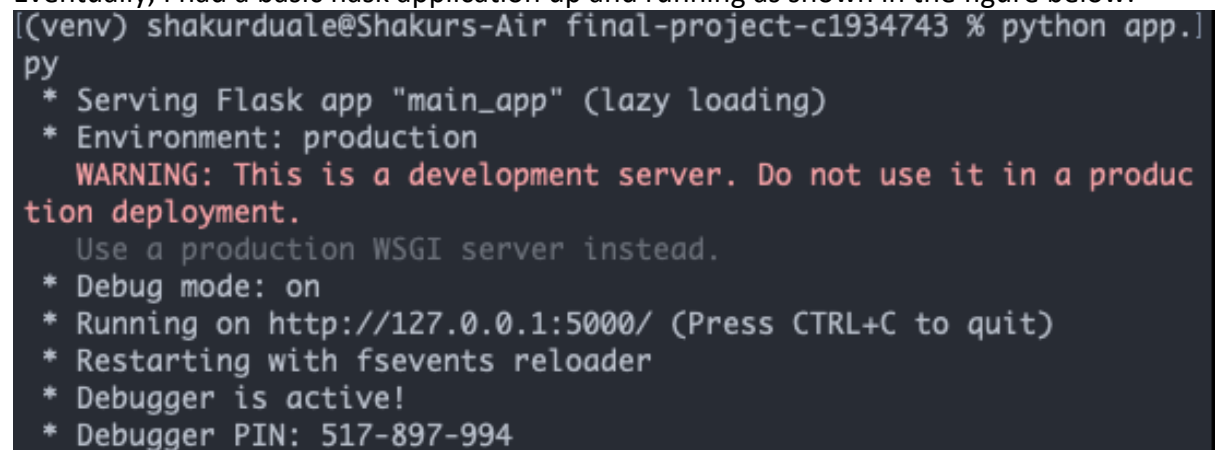
Atom

Atom is the coding environment I will be using for this project. I picked Atom because of the familiarity; it has been my go-to text editor for three years. I did not use an IDE such as PyCharm because the scale of my project is not so big that an IDE is required. Additionally, Atom can provide me with support for many different programming languages (I'll be using Python, HTML, CSS, and JavaScript at least) whereas PyCharm is built for Python. Another reason for picking Atom is that I prefer to use the command line for git control, handling virtual environments, downloading modules and libraries as well as running the application. I am not familiar enough with PyCharm or other IDEs to do this part and it will be time-consuming to learn.

Flask Setup

Flask is a web framework that can even be called a library as it is quite small. It is very suitable for smaller projects and can get a website up and running very quickly. A Flask project saves time by creating a website through HTML templates. These templates can be inherited by another HTML page to save a lot of space and time refactoring. This is very important for me considering that I do not want to waste any time on the simple design bugs. Additionally, it allows me to structure my python files and separate the main algorithms and the routes. However, the primary reason for using this web framework is that I am already familiar with it. This means that I do not need to consider other web frameworks like React, Angular and Django.

This is the first step I took in the implementation stage. I walked through the written tutorials on the [Flask website](#) to get started. This tutorial included the setup stage where you download Flask and other related dependencies as well as create a virtual environment. Eventually, I had a basic flask application up and running as shown in the figure below:

A terminal window with a dark background and light-colored text. The prompt is `[(venv) shakurduale@Shakurs-Air final-project-c1934743 % python app.]`. The output shows the Flask application starting, including a warning about development vs production mode and the server address `http://127.0.0.1:5000/`.

```
[(venv) shakurduale@Shakurs-Air final-project-c1934743 % python app.]
py
* Serving Flask app "main_app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production
  deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with fsevents reloader
* Debugger is active!
* Debugger PIN: 517-897-994
```

Figure 16: Flask application running from terminal

Figure 16 above shows what the application looks like when it's run from the terminal. As you can see from the first line we are in the project folder and have run the python file `app.py` in a virtual environment. This has started the application and hosted it on

<http://127.0.0.1:5000/> or localhost:5000. This means that the application is running locally on this computer on port 5000.

A Minimal Application

A minimal Flask application looks something like this:

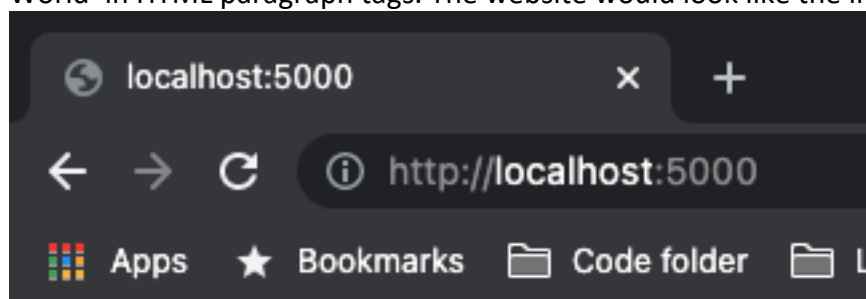
```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello_world():
    return "<p>Hello, World!</p>"
```

Figure 17: Starter Flask App – <https://flask.palletsprojects.com/en/2.0.x/quickstart/>

The flask app above is the starter code from the flask website. The first line imports the Flask library, and the next line defines the application. Flask uses decorators for routes. So, `@app.route("/")` means that the code below will be run at 'localhost:5000/' whereas `@app.route("/home")` would be run at 'localhost:5000/home'. The following line defines the function that is called when you got to that route. In that function, there is only 'Hello World' in HTML paragraph tags. The website would look like the image below:



Hello, World!

Figure 18: Flask App Starter Output

This shows the output of the starter flask code. The 'Hello World' HTML is displayed on the webpage and as you can see it is on the 'localhost:5000/' route. This is the basic setup for a Flask application.

Bootstrap Template

This project is a web application so for the front-end, I am using HTML and CSS. This project primarily focuses on functionalities produced in the backend, so I am not too focused on

creating a design. I am going to use Bootstrap CSS for most of the design and a Bootstrap Template as shown in the figure below.

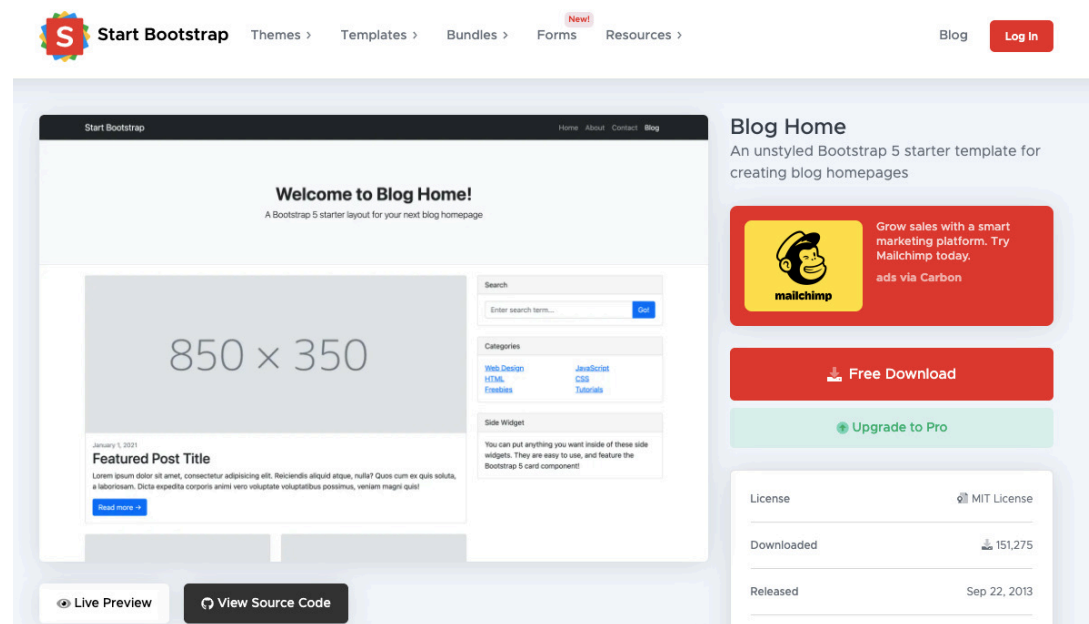


Figure 19: The Project Template - (Start Bootstrap, n.d.)

This design is useful as it gives the website a nice starting structure while also giving room for modifications. The initial plan after picking this template is to put the chart in the blog post image section, use the search bar to find a currency pair and use the categories section to pick an explanation page for the candlestick patterns.

There are two steps to implementing this template into the website. Firstly, I must download the template and move it into the application folder. Secondly, I need to refactor the HTML pages by creating a template for every page to inherit.

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
    <meta name="description" content="" />
    <meta name="author" content="" />
    <title>{% block title %}{% endblock %}</title>
    <!-- Core theme CSS (includes Bootstrap)-->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integ
  </head>
  <body class="d-flex flex-column min-vh-100">
    <!-- Responsive navbar-->
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
      <div class="container">
        <a class="navbar-brand" href="{{ url_for('what_is_a_candlestick_pattern') }}">
          <ul class="navbar-nav ms-auto mb-2 mb-lg-0">

            <li class="nav-item"><a class="nav-link" href="{{ url_for('home') }}">Home</a></li>

            <li class="nav-item"><a class="nav-link" href="{{ url_for('test') }}">Chart</a></li>

            <li class="nav-item"><a class="nav-link" href="{{ url_for('what_is_a_candlestick_pattern')
          </ul>
        </div>
      </div>
    </nav>

    {% block content %}
    {% endblock %}

```

Figure 20: Flask Template

Templating is one of the most useful functions of a web framework. With the template in Figure 20 above I can put all the repeated HTML such as the background, Header, Footer, Navbar etc and then inherit the template from another HTML page without having to type out all this HTML again. The {% block title %} is where I would put the title and the {% block content %} is where I would put the HTML for a page. This is shown in the figure below:

```

{% extends 'base.html' %}

{% block title %}
  Test
{% endblock %}

{% block content %}
  <a class="btn btn-secondary" data-toggle="button" href="{{ url_for('reset') }}">Clear Chart</a>

  <iframe src="{{ url_for('chart') }}" width="1400px" height="500px"></iframe>

  <div class="container">
  </div>

{% endblock content %}

```

Figure 21: Flask Page Inheriting Template

Figure 21 above shows just how much space is saved from using the Flask template. This saves me from having to rewrite all that HTML again. Additionally, in the future, if I need to

change the header, for example, I can do it in the template. After that, every page's header will be changed. The first line at the top of Figure 21 `{% extends 'base.html' %}` is all that is required to use a template.

I have now restructured the HTML from the bootstrap template and my website looks just like the template.

Website Design

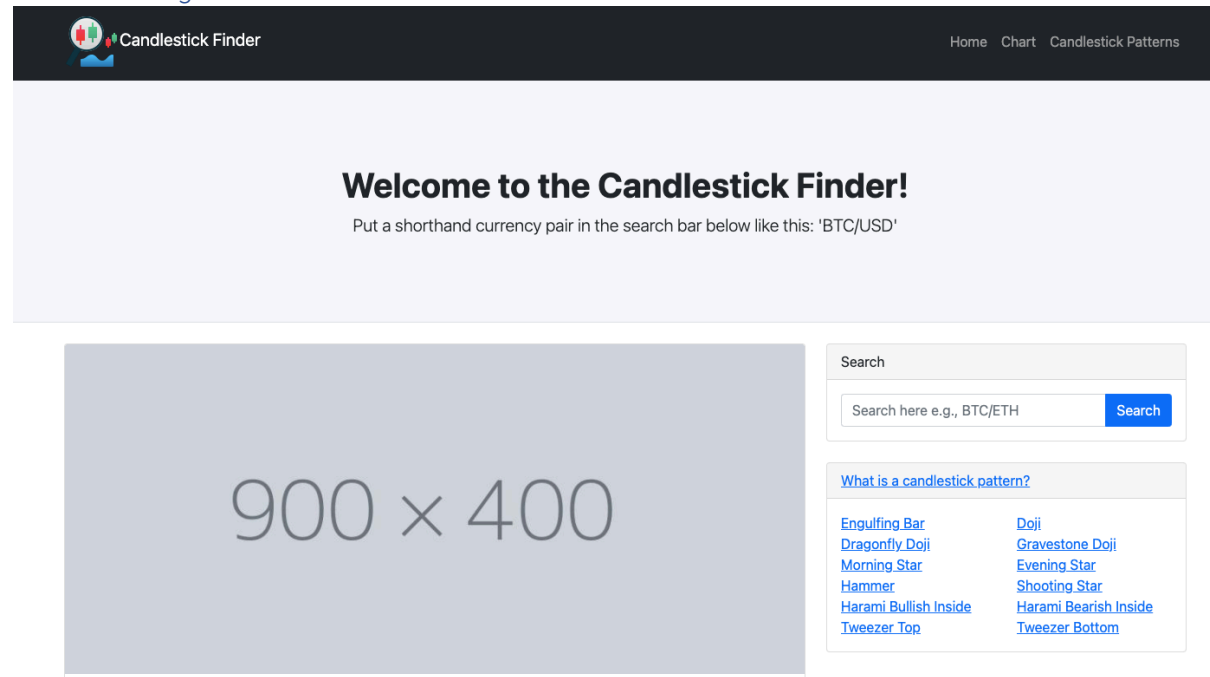


Figure 22: First Iteration - Home Page Design

Figure 22 above shows the design of the homepage. The template only needed to be tweaked slightly to create this. From this page, you would use the search bar to get to the chart page. Below the search bar are multiple links to each of the candlestick pattern's explanation pages. This initial design is simple and makes it clear how to access each of the functionalities. The design for the chart page and explanation pages will be completed in later iterations.

Version Control

Version control, also known as source control, is the practice of tracking and managing changes to software code (Atlassian, n.d.). One notable version control system is Git, and it is the one I will be using for this project. The repository will be on my GitHub account. This is extremely important as it will allow me to revert my code to a previous version if I run into too many bugs or the application crashes and I do not know how to fix it. Git has many other features such as branching but I will only be using it to save previous versions.

Project Layout

This section briefly explains the file structure of my project. The file structure that I have planned for this project is shown in the figure below.

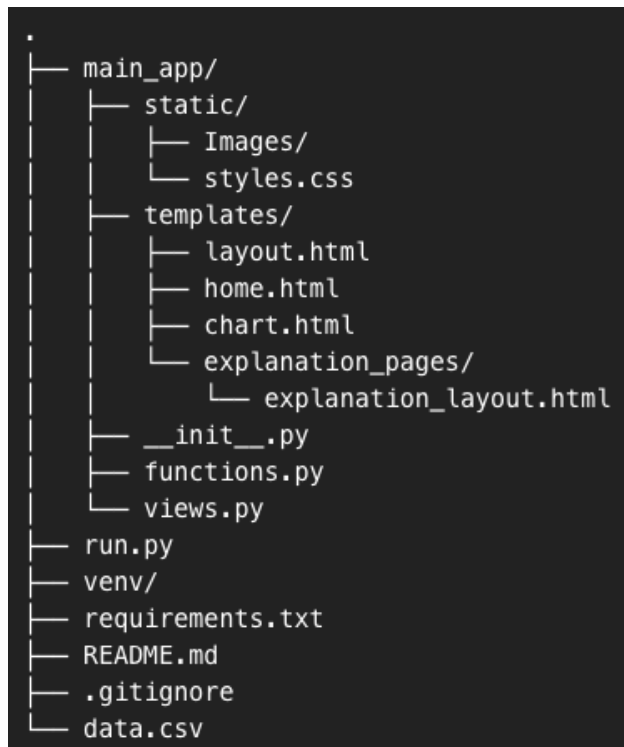


Figure 23: File Structure of my Project

Python files

In Figure 23 above, you can see that the actual application is in the `main_app` folder. This folder is a python package as you can see it contains an `__init__.py` file. This file will only contain necessary imports and a few lines of code to create the app. `functions.py` is the file that will contain the algorithms for calculating the candlestick patterns from the `data.csv` file. The `views.py` file will contain the routes for each page of the website. The `run.py` file outside of the `main_app` folder is the file that is run in the terminal to start the application.

Templates and CSS

In the templates folder, there is the `layout.html` file which is the website template. The `home.html` page is the homepage and the `chart.html` page is the chart page. The `explanation_pages` folder contains the pages for the candlestick pattern explanations. The Static folder contains an `Images` folder and the CSS file for the design of the website. This file will only include some extra design, most of the CSS will come from Bootstrap.

Extra/Other Files

There is also the `venv` folder which is for the virtual environment. The `requirements.txt` file contains a list of the dependencies for the project. The `README.md` file contains a description of the project that is shown on GitHub's repo page. The `.gitignore` file contains the names of the files that should not be tracked by Git such as the virtual environments. The `data.csv` file contains the data for the chart.

First Iteration

For the first iteration of this project, I am attempting to complete the first use case which is to get the chart of the chosen currency pair which for now will be BTC/USDT. I will not be

completing the entire use case, only the part where the chart is displayed with data obtained from the API.

Design

Twelvedata

Twelvedata (TwelveData, 2020) is the API that I will use to obtain financial data. The first reason that I'm going to use Twelvedata is that it has a free version which is useful for this project that is not being commercialised. Additionally, the API allows access to multiple financial markets including Forex, Cryptocurrency, and some stock exchanges. Thirdly twelvedata has a very convenient python library that allows me to retrieve the data and format it in multiple forms such as a list, dictionary, csv file or Pandas DataFrame. Other options were considered such as Yahoo Finance API, but this API was deprecated and there is only an unofficial version remaining.

Plotly

Plotly (Plotly, n.d.) is the python library that I am going to use to display the candlestick charts to the user. I am using this library as it provides an interactive chart that allows the user to zoom in and out, box select an area of the chart, hover over the candlesticks to see the high, low, open, and close prices as well as download the image as a png file. Additionally, you can add captions to the chart which can be modified to produce the effect of identifying candlestick patterns on the chart later. Thirdly, the data displayed in the chart gets its data through the Pandas function shown below:

```
import plotly.graph_objects as go
import pandas as pd

df = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/finance-charts-apple.csv')
```

Figure 24: Acquiring data in Plotly (Plotly, n.d.)

This complements the twelvedata API quite well as that API can acquire data in csv form. I did not consider any alternative versions as this was by far the best python library I could find for this job.

Development

Setting up Twelvedata

There is a simple tutorial on how to use the Twelvedata library on their GitHub repo's README.md file (TwelveData, 2020). This tutorial shows how to install the library, the supported parameters and a basic application shown in the figure below.

```

from twelvedata import TDClient

# Initialize client - apikey parameter is required
td = TDClient(apikey="YOUR_API_KEY_HERE")

# Construct the necessary time series
ts = td.time_series(
    symbol="AAPL",
    interval="1min",
    outputsize=10,
    timezone="America/New_York",
)

# Returns pandas.DataFrame
ts.as_pandas()

```

Figure 25: Twelvedata Example Application

Figure 25 shows how to use twelvedata. The time series constructed uses the symbol parameter to pick the stock, currency, or currency pair; the interval for the time; the output size for the number of candlesticks on the chart and the time zone is self-explanatory. The final line returns the data as a dataframe. The figure below shows how I have modified this code for my application.

```

from twelvedata import TDClient

# Initialize client - apikey parameter is required
td = TDClient(apikey="YOUR_API_KEY_HERE")

# Construct the necessary time series
ts = td.time_series(
    symbol="BTC/USD",
    interval="1day",
    outputsize=80,
    timezone="America/New_York",
)

# Returns pandas.DataFrame
ts.as_csv()

```

Figure 26: Modified Example Twelvedata Application

Figure 26 shows the code that I have modified for my application. The symbol and interval will be variable in the future but for now, I have made them constant. I have returned the data as a csv file as it is not formatted correctly when converted to a Pandas dataframe. As a csv file, it will be easy to connect to the Plotly code as Plotly has a readcsv() function. The csv file produced is shown in the figure below.


```
datetime,open,high,low,close
2022-04-05,45516.76172,45516.76172,43913.42969,44060.48047
2022-04-04,46613.35938,47193.80859,45383.82031,45491.60938
2022-04-03,46415.01953,46891.98828,45140.28906,46622.48828
2022-04-02,45812.14062,47439.01953,45561.12891,46411.10156
2022-04-01,46307.39844,47165.53906,45650.87891,45812.14062
2022-03-31,45528.14844,46709.78906,44245.00000,46307.39844
2022-03-30,47077.17969,47609.39844,45236.37891,45508.87109
2022-03-29,47460.33984,47715.64062,46618.48828,47077.17969
2022-03-28,47190.78125,48117.87891,46991.89062,47445.96875
2022-03-27,46868.57031,48210.10938,46673.83984,47118.23828
2022-03-26,44548.16016,46936.73047,44450.98047,46869.28906
2022-03-25,44332.26172,44813.17969,44110.32031,44553.94922
2022-03-24,44008.92188,45110.01172,43633.30859,44332.26172
2022-03-23,42901.32812,44239.62891,42688.44141,44008.92188
2022-03-22,42376.87891,43000.42188,41808.21875,42914.35156
2022-03-21,41016.51172,43309.62891,40897.32031,42376.87891
```

Figure 27: CSV file of data

Figure 27 above shows the data in the csv file. Every line is a candlestick, and the headers are shown in the top row. When connected to Plotly it will produce a candlestick chart with the latest prices of the BTC/USDT currency pair.

Setting up Plotly

```
import plotly.graph_objects as go

import pandas as pd
from datetime import datetime

df = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/finance-charts-apple.csv')

fig = go.Figure(data=[go.Candlestick(x=df['Date'],
                                     open=df['AAPL.Open'],
                                     high=df['AAPL.High'],
                                     low=df['AAPL.Low'],
                                     close=df['AAPL.Close'])])

fig.show()
```



Figure 28: Example from (Plotly, n.d.)

Figure 28 shows an example of a Plotly candlestick chart obtained from (Plotly, n.d.). The only other dependency required for Plotly is the pandas library. Plotly reads a csv file and converts it into a Pandas dataframe. From that dataframe, the Figure is created.

```
58 @app.route("/chart")
59 def chart():
60     fig = go.Figure(data=[go.Candlestick(x=df['datetime'],
61                                         open=df['open'],
62                                         high=df['high'],
63                                         low=df['low'],
64                                         close=df['close'])])
65     fig.write_html("/Users/shakurdual/OneDrive - Cardiff University")
66     return render_template('chart.html')
```

Figure 29: First Iteration - Plotting the chart

Figure 29 above shows the function that plots the candlestick chart. The first part creates the variable fig which is assigned to Plotly's method to create a candlestick chart. As you can see the data from the variable df is used. On line 65 there is a write_html() function which is

used to write the data on the chart to an HTML page. This will mean that going to that page will display the chart.

Testing

Output

Part of the output is shown below.



Figure 30: First Iteration – Output

As the use case is not complete yet there is nothing else to test.

Iteration summary

This chart shows the prices of the currency pair BTC/USDT which is Bitcoin to US Dollar Tether. With this the first iteration is complete. To complete this use case, I need to add a search bar to the homepage that allows the user to choose the currency pair to be displayed on the chart. This iteration went over the allocated length. This is mostly attributed to the difficulty in picking suitable APIs.

Second Iteration

For the second iteration of this project, I will be identifying candlestick patterns on the chart. This iteration corresponds to the second use case, but I will not be completing the use case entirely. I will only produce a chart with the Doji candlestick pattern displayed.

Design

For this iteration, I will be focusing on the back-end functions. I plan to create a set of functions to identify a single candlestick pattern from the data of one candle. These functions would then be looped through another function to check all the data and return a list of identified candlestick patterns. This list would be run through the update-chart function to add the annotations. It is important to do it this way so that the functions do not become intermingled, and the code is not repeated.

Annotations

By looking through the Plotly website I have found a way to annotate the chart to show the identified candlestick patterns (Plotly, n.d.).

```
fig = go.Figure(data=[go.Candlestick(x=df['Date'],
                                     open=df['AAPL.Open'], high=df['AAPL.High'],
                                     low=df['AAPL.Low'], close=df['AAPL.Close'])
                  ])

fig.update_layout(
    title='The Great Recession',
    yaxis_title='AAPL Stock',
    shapes = [dict(
        x0='2016-12-09', x1='2016-12-09', y0=0, y1=1, xref='x', yref='paper',
        line_width=2)],
    annotations=[dict(
        x='2016-12-09', y=0.05, xref='x', yref='paper',
        showarrow=False, xanchor='left', text='Increase Period Begins')]
)

fig.show()
```



Figure 31: Second Iteration – Annotation Design

Figure 31 shows how to annotate the chart with captions. This is the annotation I'll be using to identify the candlestick patterns.

Development

The figure below shows the code for the Doji candlestick pattern.

```
127 def doji(candle, alist):
128     if round(candle[1]) == round(candle[4]):
129         alist.append(('Doji', str(candle[0])))
```

Figure 32: Third Iteration - Doji Function

The Doji function shown in Figure 32 above takes two arguments. The first argument is a list with the 5 values of a candlestick taken from the csv file (date, open, close, high, low). The second argument is a list. The Doji function simply checks if the open and close prices are

relatively close to each other. If it is, then the 'Doji' string and the date of the candlestick will be appended to the list as a tuple. The reason for this list is because this function is called from inside another function shown in the figure below. This is a very early version of the function. I will modify it later to make it spot Doji candlestick patterns more accurately.

```
116 def one_bar_pattern(df, func):
117     """
118     Identifies all candlestick patterns that have one bar e.g., Doji.
119
120     """
121     data = df.values.tolist()
122     alist = []
123     for x in data:
124         func(x, alist)
125     return alist
```

Figure 33: Third Iteration - one-bar-pattern function

The Doji function was run through the function shown in figure 33 above. This function takes two arguments. The first is a Pandas Dataframe and the second is a function. The second argument is where the Doji function would go. This function goes through each candlestick to check if it is a Doji. This function will then return a list of all the candlesticks that are Doji. The use of this function is shown in the figure below.

```
94 def update_chart(alist, df=df):
95     fig = go.Figure(data=[go.Candlestick(x=df['datetime'],
96                                         open=df['open'],
97                                         high=df['high'],
98                                         low=df['low'],
99                                         close=df['close'])])
100
101     fig.update_layout(
102         title='The Great Recession',
103         yaxis_title='AAPL Stock',
104         shapes = shape_convert(alist),
105         annotations=annotations_convert(alist)
106     )
107
108     fig.write_html(os.getcwd() + "/main_app/templates/chart.html")
```

Figure 34: Third Iteration – update-chart function

The update-chart function shown in Figure 34 above takes two arguments. The first is the list of candlesticks created in Figures 32 and 33. The second argument is a Pandas Dataframe. The second parameter takes the data from the csv file by default. Lines 95-99 have been explained previously. The update-layout function from lines 101-106 updates the chart. This adds the labels on both sides. The key parts are lines 104 and 105. On line 104, the shape-convert function is taking the list of identified candlesticks as the argument and returns a list of shapes to be displayed. This is the same for line 105 but annotations-convert returns a list of annotations instead. The importance of these two lines is shown in the output.

Testing

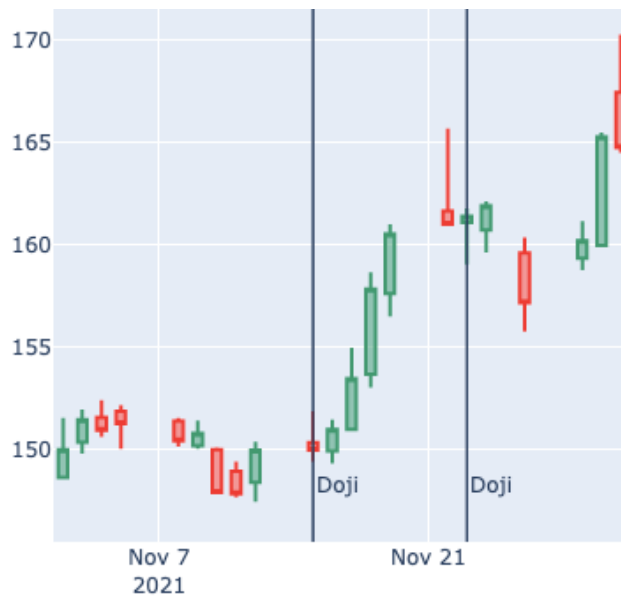


Figure 35: Third Iteration – Output

Figure 35 shows the identified Doji candlestick patterns on the chart. The black vertical lines are the shapes and the 'Doji' text at the bottom is the annotations. These are the default shapes and annotations that will be improved upon later but for now, the success criteria for this iteration has been completed. I cannot test this properly as the use case is not completed.

Iteration Summary

This iteration went over the allocated length. This was mostly because of the design of the code. I wanted to make sure that it was written well to avoid more bugs and errors down the line. Adding the annotations to the chart was much easier than expected. This was mainly due to Plotly having an example with code on their website.

Problems faced

One of the biggest problems was the algorithm for the candlestick patterns. You cannot use fixed criteria to identify candlestick patterns, they must be relative. For example, the Doji candlestick occurs when the open price equals the close price. This can be seen visually but is much harder to calculate mathematically as shown in the figure below.



Figure 36: Second Iteration - Doji Candlestick Pattern

The figure above shows a candlestick that is clearly a Doji as explained in Figure 5. However, when you look at the values, the open and close prices are not equivalent. This was solved by creating some leeway in what is identified as equivalent by rounding values. This problem shows that all the algorithms for identifying candlestick patterns will need to be tested, analysed and reviewed multiple times.

Third Iteration

For the third iteration of this project, I am going to add the search bar functionality. This should allow the user to enter the currency pair they want to see in the search bar and then be redirected to a page displaying the corresponding candlestick chart. Once this iteration is complete, the first use case will have been completed.

Design

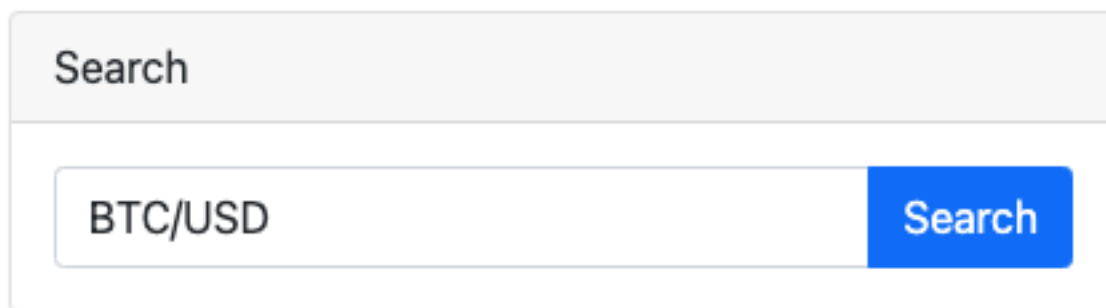


Figure 37: Second Iteration - Search Bar

Apart from the search bar, everything in the template that has already been provided will remain unchanged. Only the search bar is needed for this iteration.

Development

This is on the homepage and once it is pressed it will activate the search function as shown below.

```
<form action="{% url_for('search') %}" method="post">
  <div class="input-group">
    <input class="form-control" type="text" name="search" placeholder="Search here e.g., BTC/ETH"/>
    <input class="btn btn-primary" type="submit" value="Search">
  </div>
</form>
```

Figure 38: Second Iteration - Search Bar HTML

Figure 38 above shows the search bar inside a form. When the search button is pressed the form is submitted. When the form is submitted, the website is redirected to the 'search' route shown in the figure below.


```

76 @app.route("/search", methods=['GET', 'POST'])
77 def search():
78     if request.method == 'POST':
79         rq = request.form
80
81         type = rq.get("search")
82         # time = rq.get("time")
83         try:
84             fn.get_graph(type)
85         except:
86             flash('We Could Not Find That Currency Pair!', 'error')
87             return redirect(url_for('home'))
88
89         fig = fn.update_chart(data, df)
90         return redirect(url_for('chart'))
91
92
93 def update_chart(adict, df):
94     fig = go.Figure(data=[go.Candlestick(x=df['datetime'],
95                                         open=df['open'],
96                                         high=df['high'],
97                                         low=df['low'],
98                                         close=df['close'])],
99                  layout = dict(xaxis = dict(rangeslider = dict(visible=False))))
100
101     fig.write_html(file=f'{os.getcwd()}/main_app/templates/chart.html', include_plotlyjs='cdn')

```

Figure 39: Second Iteration - Search function

Figure 39 shows the search function. This function gets the data from the form and assigns it to the variable type. Thereafter, the `get_graph()` function from `functions.py` (written as `fn` for short when imported) is called in a try block. This function calls the Twelvedata API for the data of the currency pair given in the argument. It will then write that data to the `data.csv` file.

Testing

The output is shown in the figures below.



Figure 40: Third Iteration – Output

Figure 40 above shows the output from searching for the currency pair 'GBP/USD'. From the prices shown the correct currency pair has been displayed. Therefore, the basic flow of the first use case has been completed.

Exception Flow

If the currency pair given does not exist, then an error message will pop up and the page will be refreshed as shown below.

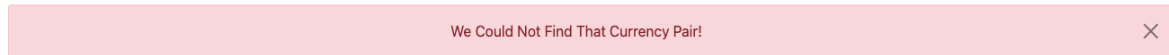


Figure 41: Second Iteration - Error Message

After the data is written to the data.csv file, the chart is updated and then the user is redirected to the chart page. This means that the exception flows of the first use case have been completed.

Iteration Summary

Through this iteration, the first use case has been completed. Although the alternative flows are not completed yet, they rely on other functionalities that have not been implemented yet so I could not do it now. This iteration was completed long before the allocated length. This is because most of the functionality for this was completed in the first iteration.

Problems faced

The first problem faced was that the time taken to redirect to the chart page and display the chart took too long (sometimes longer than 5 seconds). After looking through Plotly's documentation for a solution to this I discovered the reason for this problem as shown below.

- **include_plotlyjs** (*bool or string (default True)*) –
Specifies how the plotly.js library is included/loaded in the output div string.
If True, a script tag containing the plotly.js source code (~3MB) is included in the output. HTML files generated with this option are fully self-contained and can be used offline.
If 'cdn', a script tag that references the plotly.js CDN is included in the output. The url used is versioned to match the bundled plotly.js. HTML files generated with this option are about 3MB smaller than those generated with include_plotlyjs=True, but they require an active internet connection in order to load the plotly.js library.

Figure 42: Second Iteration - Plotly write_html documentation

Figure 42 above shows one of the parameters of Plotly's write_html() function. This parameter is True by default, and this causes the source code of the chart to be stored locally in an HTML file which is about 3MB in size. This is extremely large for an HTML file which is why it took so long to load. However, if the argument given for this parameter is 'cdn' the HTML file is stored online. This means that I won't be able to see the chart without an internet connection, however, the time taken to load the chart is much quicker (less than a second).

Fourth Iteration

For the fourth iteration of this project, I will be creating explanation pages for the candlestick patterns. This is the third use case.

Design

For this design section, I will explain the layout I chose for the explanation pages and how and where I got the information.

Research

For these explanation pages, I got most of the information from the Candlestick Bible (Raut, 2020). From this book, I took keynotes on the individual candlestick patterns. This is shown in the figure below:

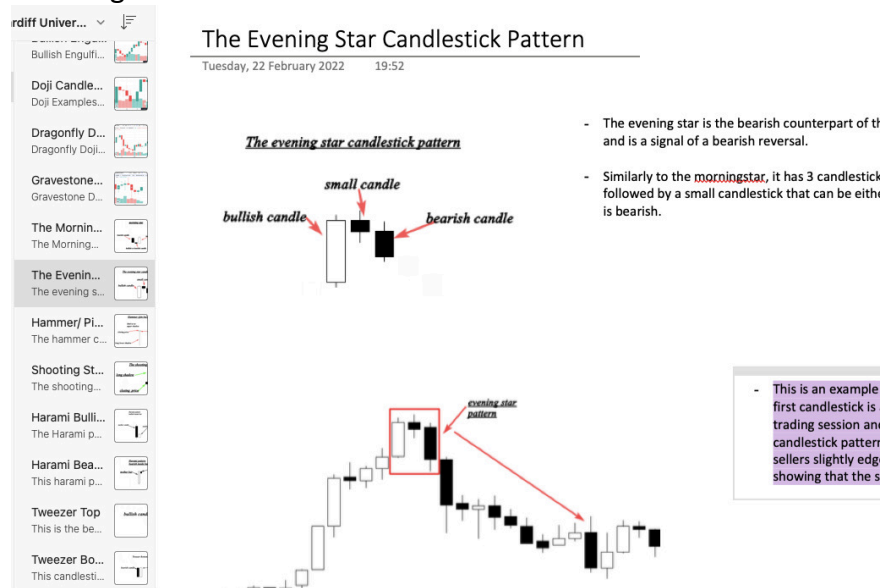


Figure 43: Fourth Iteration - Research Notes

Figure 43 above shows the notes that I took on the candlestick patterns. There were a few key points on each candlestick pattern that I wanted to note down. Firstly, I needed to have an image and explanation of what the candlestick pattern looked like. Secondly, I needed an image and an example of what the candlestick pattern looked like on a chart. Another point that is not a key point was that I wanted to show some real examples taken from live charts. These three points would then become the building blocks for the explanation pages.

Development

Creating the explanation pages

The figure below shows one of the explanation pages.

Doji

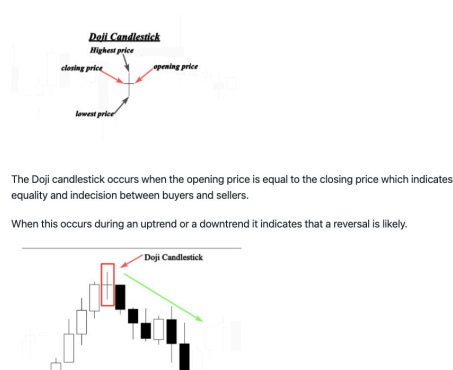
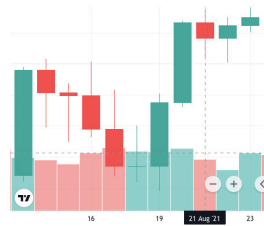


Figure 44: Fourth Iteration - Doji explanation

Doji Examples



This is a perfect example of a Doji candlestick indicating a reversal of the downtrend. Sellers tried to keep the price low but buyers brought it back up to the open. The Doji candlestick was followed by a steep uptrend.



Figure 45: Fourth Iteration - Doji Examples

Figure 44 and Figure 45 above show the explanation page for the Doji candlestick pattern. For this page, I emphasized visual learning with Images and Diagrams. The text was also deliberately kept short. This is to avoid large information dumps which would cause the user to not be as inclined to read. Common trading terminology such as ‘bullish’ and ‘bearish’ are deliberately used in the explanations to familiarise the user. With an understanding of such terminology, it will be easier to understand financial reports, predictions, and analyses.

Adding the navigation

The navigation to the candlestick pattern explanation pages was created from the categories section under the search bar in the template. This is shown in the figure below.

Search

Search

[What is a candlestick pattern?](#)

[Engulfing Bar](#)
[Dragonfly Doji](#)
[Morning Star](#)
[Hammer](#)
[Harami Bullish Inside](#)
[Tweezer Top](#)

[Doji](#)
[Gravestone Doji](#)
[Evening Star](#)
[Shooting Star](#)
[Harami Bearish Inside](#)
[Tweezer Bottom](#)

Figure 46: Fourth Iteration - Navigation to explanation pages

Testing

There is nothing to test in this section. If the links are pressed, they will take you to the explanation pages. The basic flow of the third use case is complete.

Iteration Summary

This iteration was completed much more quickly than I expected. I had allocated far more time to it beforehand. In hindsight, I did not need to allocate as much time since it was simply HTML and there was not too much to research.

Final Product Review

This iteration was the final one where I was cleaning up the code, improving the design etc. I did not need a design or testing phase for this iteration. I was only going through a list of small changes.

Home Page

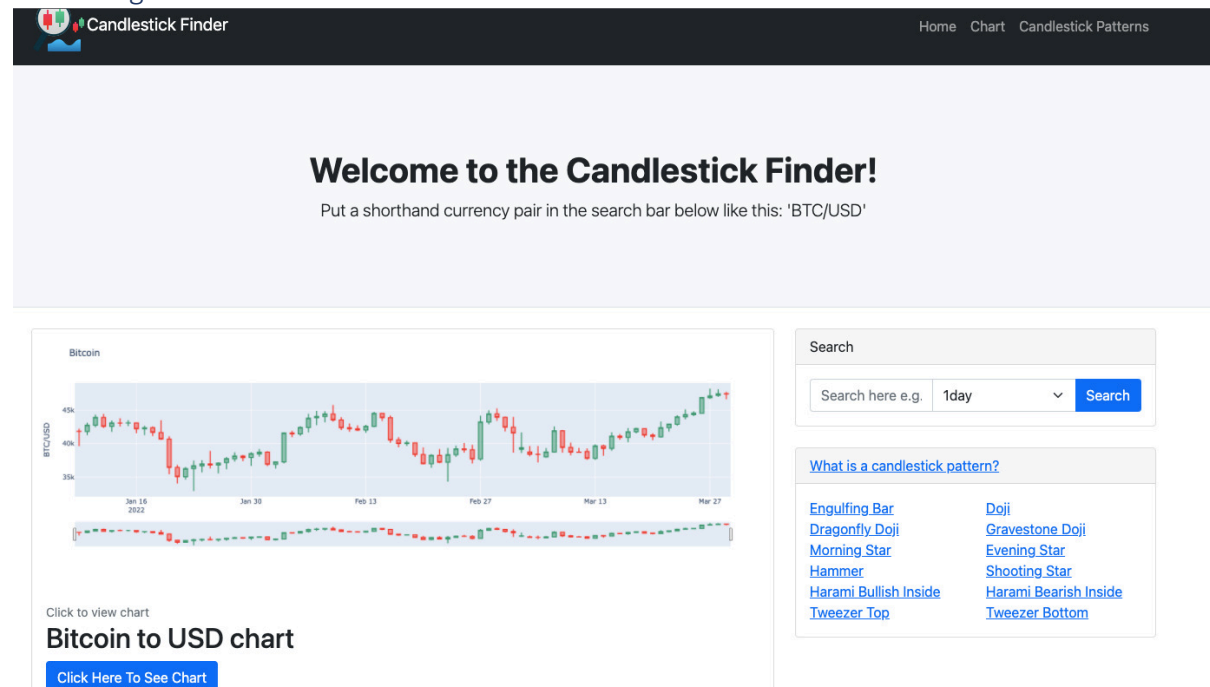


Figure 47: Homepage

Figure 47 above shows the homepage. The goal of the design was simple and clear. The search bar is easy to spot, and the user is directed to it from the header. Below the search bar are the links to the explanation pages. It may not seem obvious what they are, but it is on the chart page. I would prefer the user to first look at a chart before looking at the explanations. There is also a Bitcoin chart on the bottom left with a link to the live chart. I would have put live charts there, but it would make the website slow and they are not necessary.

Chart Page

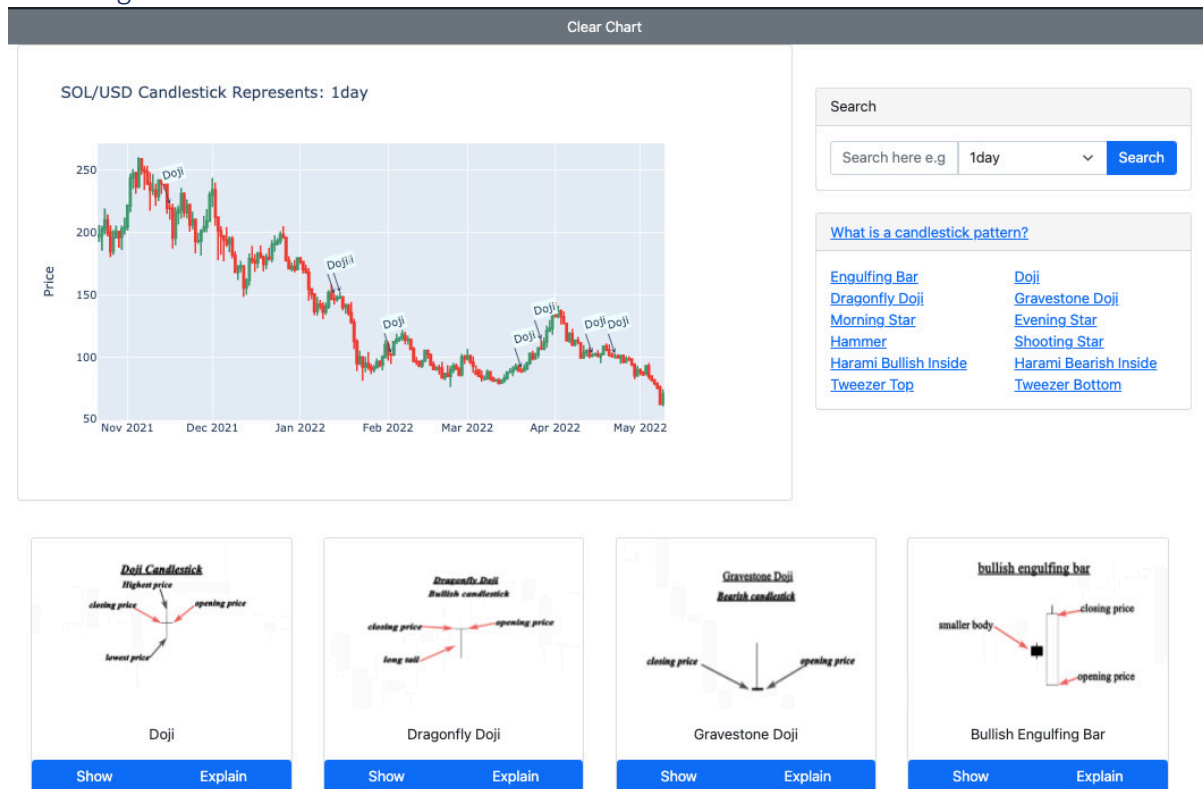
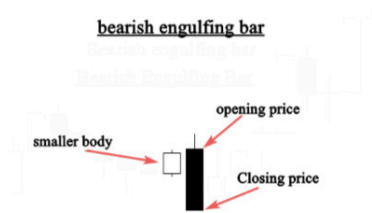


Figure 48: Chart Page

Figure 48 above shows the chart page. From this page, the user can see the live chart of the currency pair they searched for. This chart is a SOL/USD chart with each candlestick representing a 1-day trading session. You can see annotations on the chart identifying the Doji candlestick patterns. On the side is the search bar and links to explanation pages. At the bottom are cards for each candlestick pattern showing what they look like. The 'show' button will identify that candlestick pattern on the chart. If there are no instances of it, then a flash message will appear. If they have already been identified like the Doji in Figure 45, then they will be removed. The 'explain' button on the cards is a link to their respective explanation page.

Explanation Pages

Engulfing Bar



The engulfing bar pattern occurs when a candlestick fully engulfs the previous candle. It can engulf more than one but it must engulf at least one.

A true bearish engulfing bar will have the body of the candlestick fully engulf the previous candle including the shadows.



Figure 49: Engulfing Bar Explanation Page

Figure 49 above shows the explanation page for engulfing bars. Every explanation page has a format of explaining the candlestick pattern with an image; an example of the candlestick pattern with an image and potentially real-life examples explained with an image. The aim is to give all the necessary information without overwhelming the user with text. Where images can explain, words are not needed. Additionally, there is a search bar and the explanation pages section below it. Figure 46 and the other figures in this section show that the user can navigate to any page on the website from any page.

Results and Evaluation

In this section, I will be showing the results of the project in completing the requirements and use cases. The use cases will be assessed with test cases. The requirements will be assessed with simple requirement analysis. The results, the methods of testing and the original requirements will be evaluated.

Test Cases

This section shows the test cases. These test cases assess whether the use cases have been completed in the project. If the alternate flows and external flows were not that different to the basic flow or did then they will not be included.

First Test Case

The first test case assesses the basic flow of the first use case: "Get the chart of the chosen currency pair." The reason this use case was important was that it was the prerequisite to

Search

Search here e.g.

1day

▼

Search

What is a candlestick pattern?

[Engulfing Bar](#)

[Dragonfly Doji](#)

[Morning Star](#)

[Hammer](#)

[Harami Bullish Inside](#)

[Tweezer Top](#)

[Doji](#)

[Gravestone Doji](#)

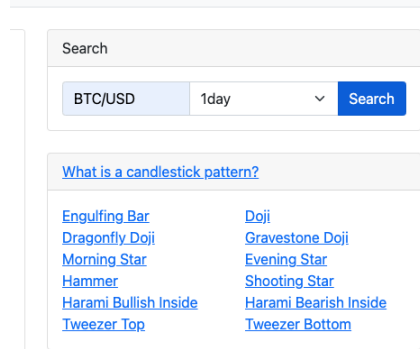
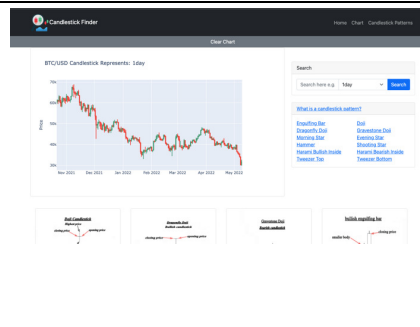
[Evening Star](#)

[Shooting Star](#)

[Harami Bearish Inside](#)

[Tweezer Bottom](#)

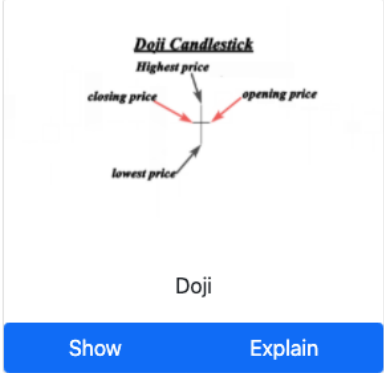

the other functionalities in the project and this is also why it had to be first. You can't identify candlestick patterns on the chart if there is no chart.

Test Case Id:1		Test Purpose: Get the chart of the chosen currency pair	
Preconditions: Start on the Homepage.			
Test Case Steps: 2			
Step No	Procedure	Response	Pass/Fail
1	Enter the currency pair chosen into the search bar e.g., BNB/USD and press the search button.		Pass
2	You have been redirected to the candlestick chart displaying the currency pair chosen in step 2.		Pass
<p>Evaluation: The test was completed successfully, and the results were as expected. Each of the alternate flows are the same but they start on a different page. Those alternate flows were completed successfully as well. This test case is quite efficient in showing that the results were as expected. However, this project only supports cryptocurrency and forex currency pairs. Therefore, if I was a user, how would I know that I should only pick a currency pair from those two exchanges? There is a lot of cryptocurrency currency pairs displayed on the homepage that hints to the user and may cause the user to choose a crypto pair. However, this is hopeful, the user should be informed clearly that these are the only supported exchanges or they might be searching for a while to no avail.</p>			

Second Test Case

The second test case evaluates the use case: "Identify Candlestick Patterns on the Chart." This use case shows the primary functionality of the application.

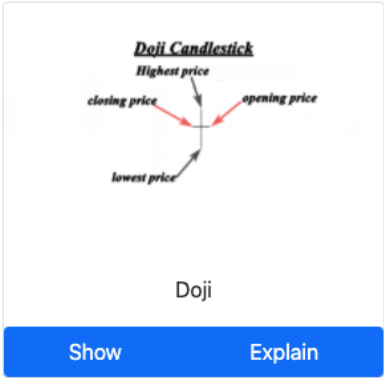

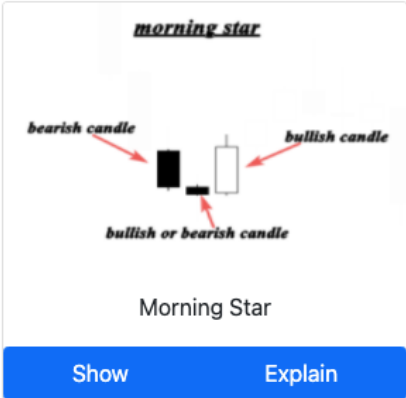

This use case shows the primary functionality of the application.	
Test Case Id:2	Test Purpose: Identify Candlestick Patterns on the Chart.
Preconditions: Start on the Chart page.	
Test Case Steps: 2	

Step No	Procedure	Response	Pass/Fail
1	Select the Doji candlestick pattern by pressing 'Show' underneath the Doji card.		Pass
2	The page has been refreshed with all instances of the Doji candlestick pattern displayed.		Pass

Evaluation: This test was completed successfully and shows that the basic flow of the second use case has been completed. This test case is completely successful in displaying the results of the application. The first step shows a card with the image of the Doji and a show button beneath it. This makes it very clear to the user what they should do. The second step also provides hard evidence that candlestick patterns are displayed on the chart. However, as you can see in the image of step 2, the labels are overlapping. If that could be fixed it would be clearer to the user and look less crowded. It is fine in this image as they are the same candlestick pattern but if it was two different ones then the user may not know what candlestick is being identified under the label.

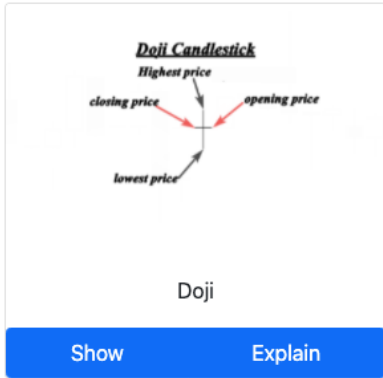



Third Test Case

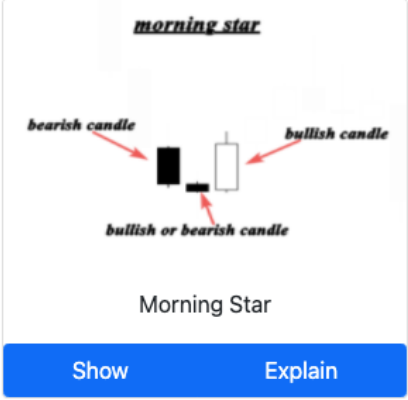

Test Case Id:3		Test Purpose: Identify Candlestick Patterns on the Chart (First Alternate flow).	
Preconditions: Start on the Chart page.			
Test Case Steps: 4			
Step No	Procedure	Response	Pass/Fail

1	Select the Doji candlestick pattern by pressing 'Show' underneath the Doji card.		Pass
2	The page has been refreshed with all instances of the Doji candlestick pattern displayed.		Pass
3	Select the Morningstar candlestick pattern.		Pass
4	All instances of both the Morningstar and Doji candlestick pattern are displayed on the chart.		Pass

Comments: This test was completed successfully and shows that multiple candlestick patterns can be identified at the same time. The first alternative flow of the second use case has been completed. This test case is the same as the second one in how well it has shown the results. The results obtained are strong and are exactly as they should be. The overlapping labels cannot be seen here but as explained in the second use case, they are a problem.

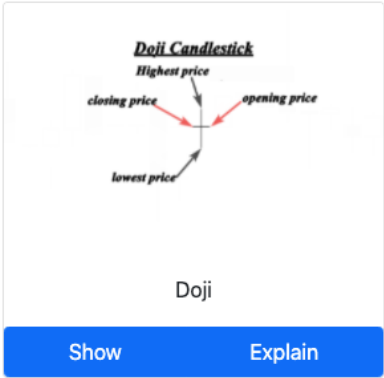
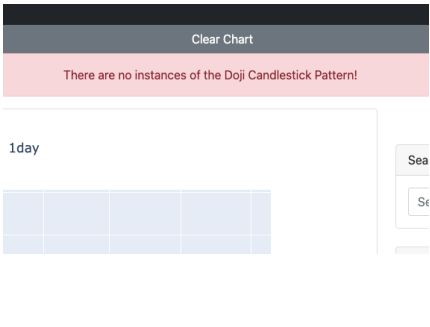
Fourth Test Case

Test Case Id:4		Test Purpose: Identify Candlestick Patterns on the Chart (Second Alternate flow).	
Preconditions: Start on the Chart page.			
Test Case Steps: 5			
Step No	Procedure	Response	Pass/Fail
1	Select the Doji candlestick pattern by pressing ‘Show’ underneath the Doji card.		Pass
2	The page has been refreshed with all instances of the Doji candlestick pattern displayed.		Pass
3	Press the clear chart button.	<div><p>The clear chart button is in grey at the top of the page.</p><p>All of the annotations and labels have been removed</p></div>	Pass

		from the chart as shown above.	
4	Select the Morningstar candlestick pattern.		Pass
5	The page has been refreshed with only instances of the Morningstar candlestick pattern displayed.		Pass
Comments: This test was completed successfully and shows that the chart can be cleared. The second alternative flow of the second use case has been completed. The clear chart button does as it is expected however, the design and placement of the button are not aesthetically pleasing which in turn would subconsciously make the user look at the website with lower expectations.			

Fifth Test Case


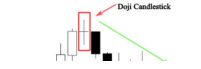
Test Case Id:5		Test Purpose: Identify Candlestick Patterns on the Chart (Exception flow).	
Preconditions: Start on the Chart page.			
Test Case Steps: 2			
Step No	Procedure	Response	Pass/Fail

1	Select the Doji candlestick pattern by pressing 'Show' underneath the Doji card.		Pass
2	The page has been refreshed and a flash message is displayed saying something like "No instances of the Doji candlestick pattern found".		Pass
Evaluation: This test was completed successfully and shows that the user is given feedback if there are no candlestick patterns are identified. The exception flow of the second use case has been completed. The flash message in step 2 gives the user clear informative feedback where interactive feedback is insufficient. However, it is debatable whether no instances of the candlestick pattern being identified can be considered an error. The choice of colour could be better.			

Sixth Test Case

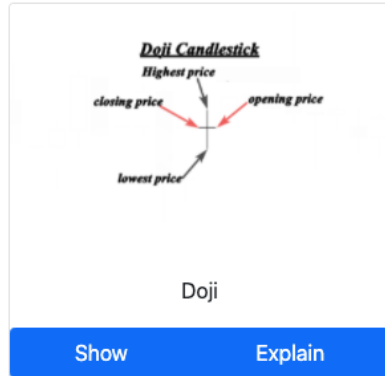
This test case corresponds to the final use case. This use case shows the secondary functionality of the application.


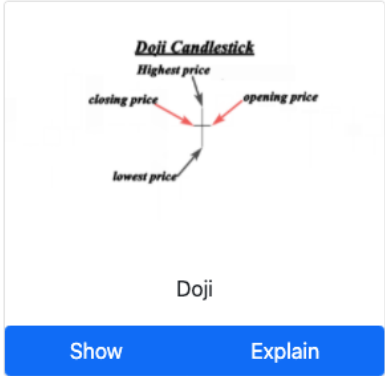

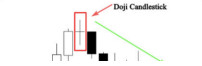
functionality of the application.			
Test Case Id:6		Test Purpose: See the explanations for the candlestick patterns	
Preconditions: Start on the Home page.			
Test Case Steps: 2			
Step No	Procedure	Response	Pass/Fail
1	Pick the Doji candlestick pattern in the section under the search bar.	<div>What is a candlestick pattern?</div> <div><div>Engulfing BarDragonfly DojiMorning StarHammerHarami Bullish InsideTweezer Top</div><div>DojiGravestone DojiEvening StarShooting StarHarami Bearish InsideTweezer Bottom</div></div>	Pass

2	You have been redirected to the explanation page of the Doji candlestick pattern.	<p>Doji</p>  <p>The Doji candlestick occurs when the opening price is equal to the closing price which indicates equality and indecision between buyers and sellers.</p> <p>When this occurs during an uptrend or a downtrend it indicates that a reversal is likely.</p> 	Pass
---	---	---	------

Evaluation: This test was completed successfully and shows how easy it is for the user to navigate to the explanation pages. In step 2 you can see the Doji explanation page. It is clear to see that images and diagrams are being used. Additionally, the amount of text is small and spaced out making it easier for the user so that reading does not feel like a chore. The first step shows the links to the explanation pages. It is questionable whether the user would press one of those links knowing that it will take them to the explanation page of the chosen candlestick pattern. Only the first link which is “What is a candlestick pattern?” gives the user a clue as to where they are going when they press the link.

Seventh Use Case

Test Case Id:7		Test Purpose: See the explanations for the candlestick patterns (Alternate flow)	
Preconditions: Start on the Chart page.			
Test Case Steps: 4			
Step No	Procedure	Response	Pass/Fail
1	Select the Doji candlestick pattern by pressing ‘Show’ underneath the Doji card.	<div><p><u>Doji Candlestick</u></p></div>	Pass

2	The page has been refreshed with all instances of the Doji candlestick pattern displayed.		Pass
3	The page has been refreshed with all instances of the Doji candlestick pattern displayed.		Pass
4	You have been redirected to the explanation page of the Doji candlestick pattern.	<p>Doji</p>  <p>The Doji candlestick occurs when the opening price is equal to the closing price which indicates equality and indecision between buyers and sellers.</p> <p>When this occurs during an uptrend or a downtrend it indicates that a reversal is likely.</p> 	Pass
Comments: This test was completed successfully and gives another option on to the user on how to get to the explanation pages. The “Explain” button in the image on step 3 seems to inform the user on what will happen when they press the button a lot more than the link shown in the 6 th test case. This is because they can see the image of the Doji candlestick pattern. If they have already pressed show, then they will have seen the identified Doji candlestick patterns on the chart which would make the decision to press the “Explain” button even more informed.			

Test Case Evaluation

In this section, I will evaluate the advantages and disadvantages of the use of test cases

Advantages (+)

- Complement the use cases well as they go step by step. This means that we can assess each step of the use case.

- Easy to calculate whether a use case has passed. A use case has passed if every step passed. If not, that use case was not completed.

Disadvantages (-)

- A lot of the test cases did not have many steps. This is because the use cases also did not have many steps. However, if the use case did not have many steps, then it is debatable whether test cases are the correct method to assess completion.
- Cannot prove the performance of an action. For example, how can we prove that it took less than 3 seconds for the Doji candlestick patterns to be displayed on the chart after a button has been pressed? This cannot be done in a test case.
- Navigating through pages cannot be assessed with test cases. At most I can show screenshots of the page but that cannot prove anything.

Test Case Summary

The disadvantages identified above raise the question of whether test cases were the appropriate way to test the use cases. However, although algorithms are used in this project, they are not complex algorithms that use advanced data structures. The hash table is the most complex data structure used in this project so unittests testing the performance of the algorithms did not seem applicable. I could have used speed tests to assess the speed of the candlestick patterns being displayed on the chart but in reality, most of the time is consumed by the page refreshing. This is out of my power and relies on the user's internet connection and is, therefore, a poor test to use. Additionally, such tests cannot assess the explanation pages. I could have used surveys or user research methods to complement unittests and this would be a great way to assess the usefulness of the explanation pages. However, most people don't have any background in trading at all and so it would be very difficult to get some veteran traders to take part in the research. This is proven by the research I completed in my design thinking module where only one person actively traded. Additionally, a multitude of testing methods such as these would be far too time-consuming. Therefore, test cases seemed to be the most practical method to use and the speed metrics can be shown in the demonstration video as proof anyway.

Requirements Analysis

For this project, I had four functional requirements and an assortment of non-functional requirements. We can assess whether these functional requirements have been completed by checking if the acceptance criteria has been met.

First functional requirement

The first functional requirement states: "The application must allow the user to choose which chart they want to analyse." The acceptance criteria is shown below.

1.) The application has a search bar that displays the chart of the currency pair put in the search bar. The currency pair will be abbreviated e.g., "GBP/USD".

Search

Search here e.g. 1day

Search

Figure 50: FR1 – First Acceptance Criteria

The figure above shows proof that the first acceptance criteria has been met.

2.) The time between the user pressing search and the corresponding chart being displayed must be within 5 seconds.

This acceptance criteria has been tested and passed in the demonstration video provided with this report.

3.) The application must allow the user to pick currency pairs from at least one financial exchange market e.g., forex, cryptocurrency.

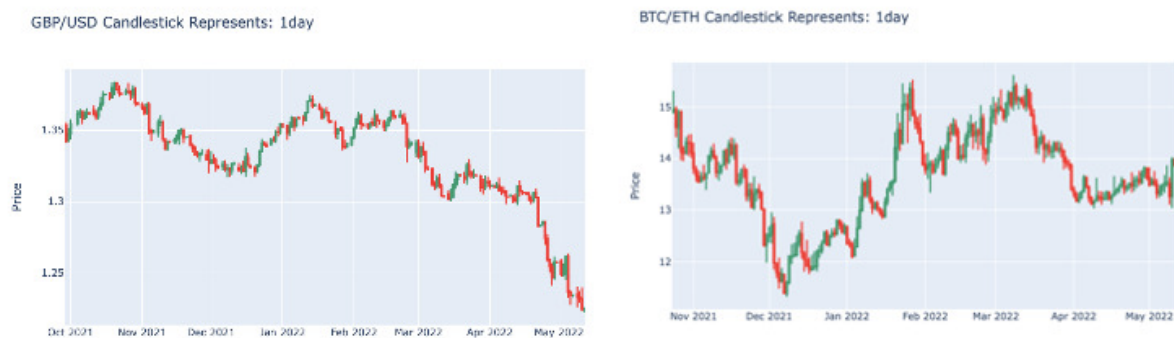


Figure 51: FR1 - Third Acceptance Criteria

The figure above clearly shows a forex chart and a cryptocurrency chart. This shows that the acceptance criteria has been exceeded.

In summary, the first functional requirement has been met perfectly.

Second Functional Requirement

The second functional requirement states: “The application must display a candlestick chart to the user”. The acceptance criteria is shown below.

1.) The candlestick chart displayed must be interactive. It is not enough for the chart to be a still image.



Figure 52: FR2 - First Acceptance Criteria

The figure above shows that the candlesticks can be highlighted to see the open, close, high, and low prices. Additionally in the top right corner from the left is a camera icon that allows the user to download the chart as a png. A magnifying glass icon to zoom in or out. A + sign with arrows to pan across the chart. An outline of a square for box select. This is the feature I used in the figure above to zoom into the chart. A plus sign to zoom in and a minus sign to zoom out. A cross symbol to auto-scale. A Home symbol to reset axes and finally chart symbol which when the mouse hovers over it, shows the version of Plotly being used. The figure above provides clear evidence that the acceptance criteria has been met.

2.) The chart must have clear axes with the price on the left and the time on the bottom. The chart may also contain titles and labels.

This acceptance criteria has been met as shown in figure 52.

In summary, the second functional requirement has been completed perfectly.

Third Functional Requirement

The third functional requirement states: "The application must allow the user to see candlestick patterns on the chart". The acceptance criteria is shown below.

1.) The candlestick patterns must be identified clearly on the chart with accurate labels.



Figure 53: FR3 - First Acceptance Criteria

The figure above shows that the candlestick patterns have been identified clearly with labels that are simple and do not overly obstruct the chart. This is evidence that the first acceptance criteria has been met.

2.) The candlestick patterns identified must be consistent with what the candlestick pattern looks like.



Figure 54: FR3 - Second Acceptance Criteria

The figure above shows the Doji candlestick pattern. This is also the candlestick pattern that has been identified in figure 53. As you can see, the identified candlestick patterns are consistent with the figure above. This is even in the case where figure 53 is zoomed in. If figure 53 was not zoomed in, then it would be even more consistent with the Doji candlestick pattern. This is evidence that the second acceptance criteria has been met.

3.) The user should be able to choose which candlestick patterns to display and hide the ones they no longer want to see on the chart.

This acceptance criteria has been met. The evidence is in the demonstration video that is provided with this report.

In summary, the third functional requirement has been completed to perfection.

Fourth Functional Requirement

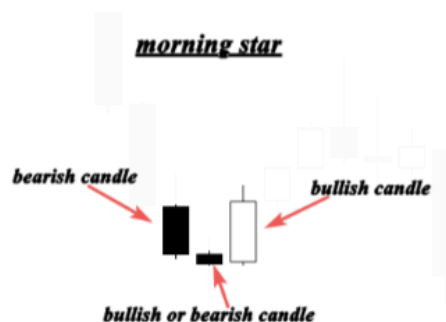
The fourth functional requirement states: “The application must allow the user to see the definitions and explanations for the candlestick patterns.” The acceptance criteria is shown below.

1.) The explanation pages can be accessed from any page.

The final product review provides evidence that this acceptance criteria has been met.

2.) Each explanation page must show clearly what the candlestick pattern looks like.

Morning Star



The Morning Star is a bullish reversal signal. It has 3 candles and the first one is a bearish candle with a large body, the second is a small candle that can be either bearish or bullish. The third is a bullish candle that closes higher than the open of the first candle.

Figure 55: FR4 - Second Acceptance Criteria

The figure above shows the Morningstar candlestick pattern on the Morningstar explanation page. Every other explanation page similarly shows what the candlestick pattern looks like. This shows that the acceptance criteria has been met.

3.) Each explanation page must explain clearly what the candlestick pattern means.

This acceptance criteria has been met. Every explanation page explains what the candlestick patterns mean as shown in figure 55.

4.) Each explanation page should contain examples.



This is an example of the tweezer bottom at the bottom of a downtrend. This shows that sellers pushed the price down significantly in the first session but the buyers dominated the next trading session and brought the price back up to the open of the first trading session.

Figure 56: FR4 - Fourth Acceptance Criteria

The figure above shows an example of the Tweezer Bottom candlestick pattern. Every other candlestick pattern explanation page also contains at least one example. This acceptance criteria has been met.

5.) Each explanation page could contain real-life examples.

Bearish Engulfing Bar Examples



There was an upward trend that started at the beginning of September. However, on the 7th of September there was a huge bearish engulfing bar that consumed the previous 5/6 candles and triggered a trend reversal.

Figure 57: FR4 - Fifth Acceptance Criteria

The figure above shows a real-life example of the bearish engulfing bar taken from crypto.com. Not all of the candlestick pattern explanation pages contain real-life examples. Therefore, this acceptance criteria has not been fully achieved. However, the criteria states that each page “could” include real-life examples, so I have not failed by coming short.

In summary, this functional requirement has been completed but not to perfection.

Requirements Evaluation

Although the methods of testing are extremely limited and basic, in this case, that is okay. The criteria for the requirements were the same. Even though that is the case, I would not change them as making them more elaborate is pointless. The criteria given are enough to test whether the requirements have been completed.

Heuristic Evaluation

In this section, I will be evaluating my usage of the 10 heuristics in this project (Nielsen, 1994).

Visibility of System Status

According to Nielsen's 10 general principles, Visibility of System Status means that the design should always keep users informed about what is going on, through appropriate feedback within a reasonable amount of time (Nielsen, 1994). Throughout the entirety of this project, I believe this heuristic has been satisfied. Through multiple methods, I have ensured that the speed of the application is fast.

- **include_plotlyjs** (*bool or string (default True)*) –
Specifies how the plotly.js library is included/loaded in the output div string.
If True, a script tag containing the plotly.js source code (~3MB) is included in the output. HTML files generated with this option are fully self-contained and can be used offline.
If 'cdn', a script tag that references the plotly.js CDN is included in the output. The url used is versioned to match the bundled plotly.js. HTML files generated with this option are about 3MB smaller than those generated with include_plotlyjs=True, but they require an active internet connection in order to load the plotly.js library.

Figure 58: Second Iteration - Plotly write_html documentation

This example of using 'cdn' to reduce the loading speed of the charts has been explained in the third iteration summary of the implementation section. However, this was not the only case of using 'cdn' within this project.

```
10 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integ
```

Figure 59: Bootstrap cdn link

The figure above shows a cdn link to a bootstrap file online. At the start I had this file downloaded and placed inside the static folder of my project. This file contained over 10,000 lines of CSS which was very heavy. As a result, the pages took a while to load. When I replaced the Bootstrap file with this "cdn" link, I could visibly see the improvement in the website's performance. If I press a link now, then I would be taken to the corresponding page almost instantaneously. I believe that fast speeds are the best form of feedback. Additionally, the entire Bootstrap contains simple effects such as a slight colour change when the user hovers over a button for example. These simple effects give subtle feedback to the user to let them know the website is working.

Match between system and the real world

According to Nielsen's 10 general principles, a match between the system and the real world means that the design should speak the users' language. Use words, phrases, and concepts familiar to the user, rather than internal jargon. Follow real-world conventions, making information appear in a natural and logical order (Nielsen, 1994). It can be seen through my final product review that I have not overwhelmed the user with information and the information given is inclusive and not filled with jargon. However, as explained in my user interface design, the explanation pages need to have jargon to teach beginner traders the language. This is best explained in the figure below.



The Doji candlestick occurs when the opening price is equal to the closing price which indicates equality and indecision between buyers and sellers.

When this occurs during an uptrend or a downtrend it indicates that a reversal is likely.

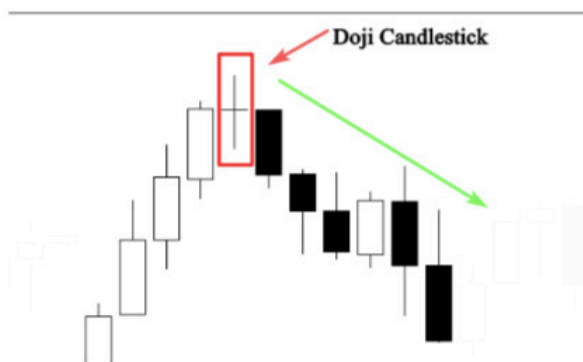


Figure 60: Doji Example – Heuristics

The figure above shows the explanation for the Doji candlestick pattern. The final sentence reads: "When this occurs during an uptrend or a downtrend it indicates that a reversal is likely." Uptrend and downtrend are jargon terms, but it is obvious what they mean when you look at the chart at the bottom. A reversal is also quite self-explanatory given the context. However, if I said it would indicate a bearish reversal that would be too much. This light use of jargon is important in getting beginner traders used to the language they will encounter day-to-day if they take trading seriously.

User control and freedom

Users often perform actions by mistake. They need a clearly marked "emergency exit" to leave the unwanted action without having to go through an extended process (Nielsen,

1994). From my Final Product Review section in the implementation, it can be seen that the user can navigate to any page on the website from their current page. Just as I explained in the user interface design section, this is the highest level of freedom the user can have over the website. They will never have to take two steps to get to where they want to go. Every figure in the Final Product Review shows the search bar and the links to the explanation pages as well as the navbar at the top. This gives the user a sense of control through familiarity.

Consistency and standards

It is essential for the application to have consistency and standards. Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform and industry conventions (Nielsen, 1994).

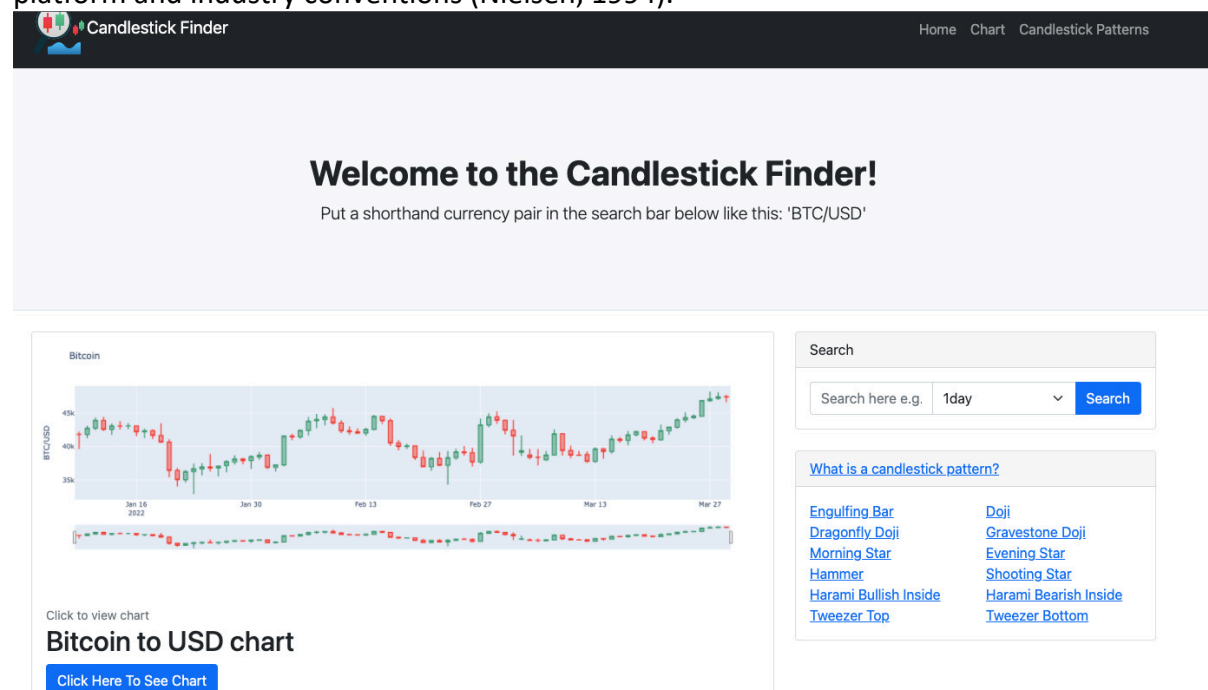


Figure 61: Homepage- Heuristics

The figure above shows the homepage of the application. The search bar on the right is the typical look for a search bar and says the word “search” in the header, hint text and submit button. This conforms to industry standards just like the navbar at the top. Although the rest of the website may look unfamiliar or unique, I have ensured that the functionalities that provide the user with navigation through the website conform with industry norms. This gives the user a sense of familiarity and control immediately upon reaching the website.

Error Prevention

Good error messages are important, but the best designs carefully prevent problems from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action (Nielsen, 1994). The Final product review section shows that there is hint text in the search bar, instructions in sub-headings and other similar hint text that give the user instructions in case the action to take is confusing. The project is quite small in size with only two primary functionalities so there was no need to excessively focus on error prevention. Keeping the website simple was

the best error prevention. There is no extra information or large info dumps anywhere on the website.

Recognition rather than recall

Minimize the user's memory load by making elements, actions, and options visible. The user should not have to remember information from one part of the interface to another. Information required to use the design (e.g., field labels or menu items) should be visible or easily retrievable when needed (Nielsen, 1994). As explained in the user interface design section, the simplicity and size of the website made this heuristic very easy to meet. The final product review shows that the user can navigate to any page with 1 click. Therefore, there is no navigation process for the user to recall.

Flexibility and efficiency of use

Shortcuts — hidden from novice users — may speed up the interaction for the expert user such that the design can cater to both inexperienced and experienced users. Allow users to tailor frequent actions (Nielsen, 1994). I did mention in the user interface design section that this heuristic is not relevant for my application due to the small scale. I was unable to add accelerators, and personalisation features to this project. The only feature that could be considered a customisation feature is shown below.

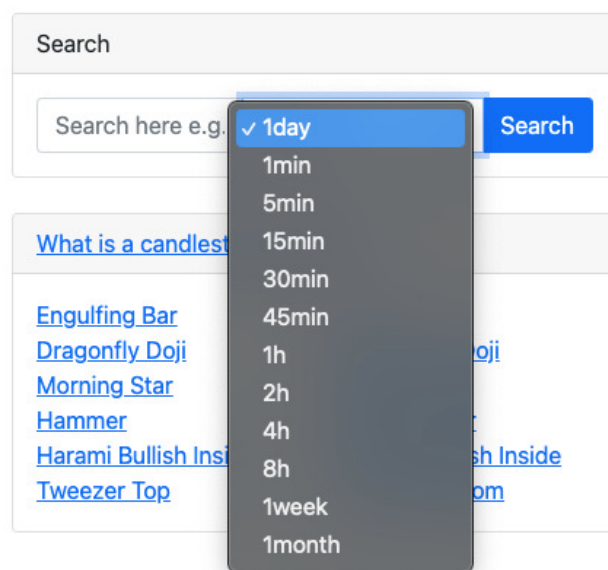


Figure 62: Timeframe – Heuristics

The figure above shows that the user can search for a currency pair while also being able to customise which timeframe they would like to choose. If the user picks "30min" then each candlestick on the corresponding chart will show a 30-minute trading session. This is the only customisation feature I have included in the application. Therefore, I can only say that I have touched on this heuristic and have not fully implemented it into my application.

Aesthetic and minimalist design

Interfaces should not contain information which is irrelevant or rarely needed. Every extra unit of information in an interface competes with the relevant units of information and diminishes their relative visibility (Nielsen, 1994). Throughout my design, prototypes, and implementation, I have followed this heuristic closely. In my final product review, you can

see that I have not added any redundant features or overloaded the user interface with information. For my application to provide the user with as much value as possible, I needed to keep a minimalist design. The main reason for this is that part of my target audience is beginner traders. The one thing that overwhelms beginners is too much information. That is why I tried to keep everything nice and simple. This is also good for veteran traders who use the website as they can access the chart they want to see quickly and easily. If I had a multitude of features for my application this type of design may not have been optimal but for my project that was not the case.

Help users recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (no error codes), precisely indicate the problem, and constructively suggest a solution (Nielsen, 1994). The figures below show the error messages in my application.

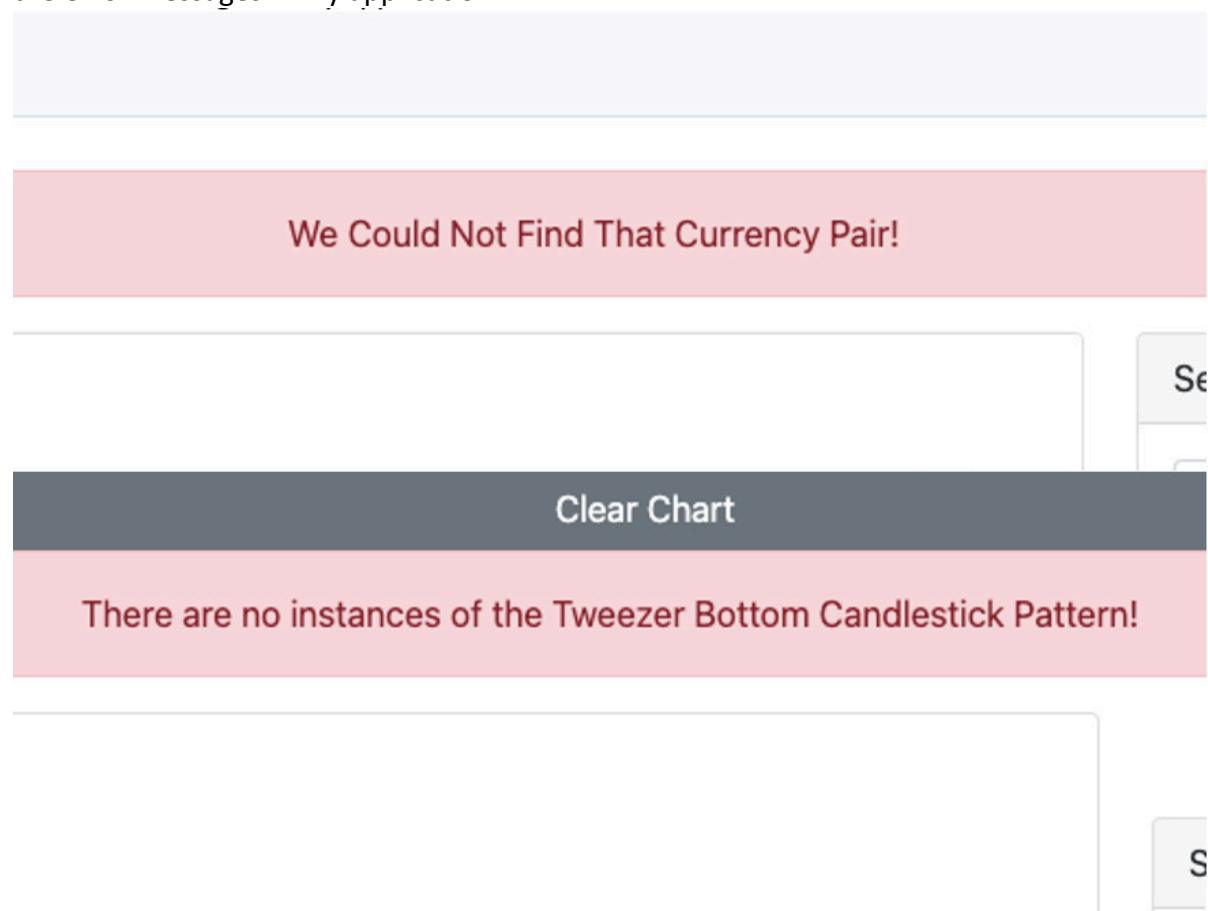


Figure 63: Error Messages - Heuristics

The figure above shows the error messages in my application. These error messages were only used when there was no interactive feedback method applicable. The first error message occurs when the user misspells the currency pair they are looking for, searches for one that does not exist or searches for one that the application does not support. As explained in the user interface design section, no solutions were provided for these error messages as they are quite self-explanatory in the sense that identifying the problem provides the solution.

Help and documentation

It's best if the system doesn't need any additional explanation. However, it may be necessary to provide documentation to help users understand how to complete their tasks (Nielsen, 1994). As explained in the user interface design, this heuristic was not necessary for the scope of my project. Therefore, there was no help and documentation implemented.

Results and Evaluation Summary

The test cases and requirements analysis both show that the project's aims and objectives were achieved. However, the usability of the website and the user interface design were not tested. Although the heuristics were evaluated, there was no actual testing done to assess the user interface design. Even though I did not prioritise the design and look of the website it is not something that should be neglected. However, if I completed user research and let people test my application and give feedback then the potential results could have been far more than what I currently have. It would also give a more realistic evaluation of the usefulness of the application. Users would likely have given feedback on the design and functionalities that I never would have thought of. It is a shame that this was not possible due to time constraints.

Future Work

The application I have produced is one that was tailored to the scope of this final year project. Perhaps if I had more time there would be more features implemented or the current features would reach a higher level of quality. Maybe this application could have been deployed online for the public to use. This section details what I would implement if I was given more time in the future.

Add More Candlestick Patterns

I have included a lot of the common and primary candlestick patterns, however, there are still many more that I have yet to implement. If I had more time, I would try my best to implement as many as possible. This would increase the value of the product a lot more for veteran traders in contrast to beginner traders.

More Examples in Explanation Pages

I believe that the more examples included on the explanation page the better. This is so the user can go through the examples and build familiarity with the candlestick pattern while also becoming more used to the trading jargon. This is especially useful to beginner traders. In the future, I would focus on this.

Improved Performance

When I created the algorithms, chose the APIs, and designed the routes and functions, I only focused on performance to a superficial level. For example, displaying the chart within 3 seconds of having searched for it. In the future, I would like to improve performance with a focus on the Big O Notation.

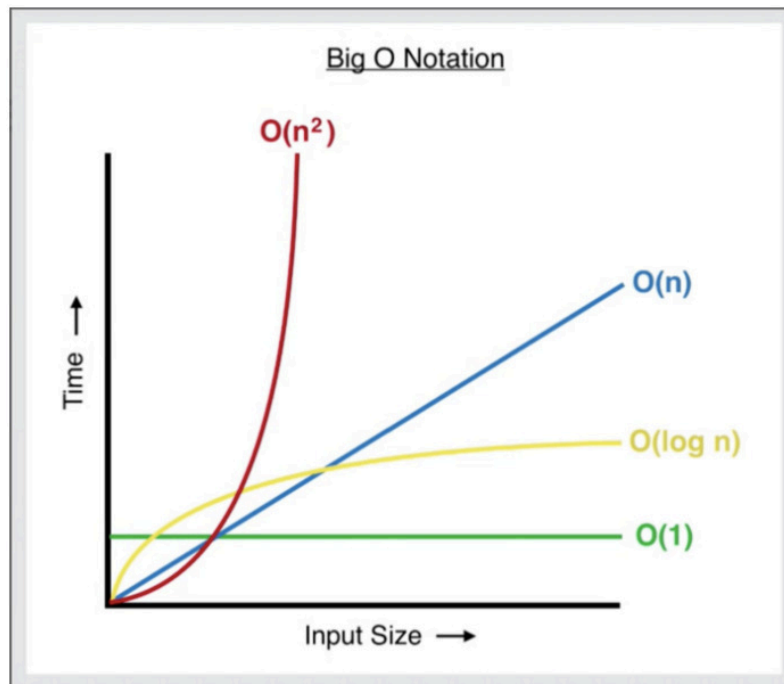


Figure 64: Big (O) notation (Croy, 2020)

The figure above shows some different levels of performance in Big O Notation. In Big O Notation $O(1)$ refers to constant time. This means that no matter how big the input size is the time taken is constant. This is the best time in Big O notation but is rarely possible. There is also linear time which is $O(n)$. This means that as input size increases, time increases. This time is not good at all for large datasets. The second-best time $O(\log(n))$. This is also called logarithmic time. Logarithmic time is best explained with the example of flicking through a book. If linear time is flipping every page one by one to get to the page you want, then logarithmic time is going straight to the middle page and if the page you want is after this page you would go to the middle of the second half and so on. This time is optimal and is what I would want my application to reach.

JavaScript

In my application, if the user selects a candlestick pattern to display on the chart the whole page is refreshed. If they decide to clear the chart or add a second candlestick pattern to the chart it will also be refreshed. This is quite inefficient and does not look nice on the website. In the future, I would redo these functionalities in JavaScript as this would allow the chart to refresh on the page without the whole page having to refresh which is very inefficient. The most important feature that it would enable is changing the timeframe within the chart without having to search for the chart again with a different timeframe.

Support and Resistance Calculations

Support and Resistance Levels are another fundamental tool in the analysis of financial charts (Murphy, 2021). Candlestick patterns can show you what the next trend could potentially be. However, support and resistance allow you to calculate when the next trend will end.



Figure 65: Support and Resistance - (Murphy, 2021)

The figure above shows resistance levels which act as a ceiling price. The resistance and support levels are calculated through trends in past data. This gives financial traders assurance with fluctuating prices. For example, if the price is dropping rapidly, you would not panic unless it falls too far below the support level. After that, it is best to sell as the market is crashing. In the future, this would be a great feature to implement that would complement the identified candlesticks very well.

Conclusions

In this project, the primary aim was to create an application that analyses the data of a financial chart and compares the high, low, open, and close prices of each trading session to calculate and identify the candlestick patterns on the chart and visualise these data for the user. The secondary aim was to provide the user with explanations for each of the candlestick patterns used in the application to learn from.

The final product succeeded in meeting the primary aim. It identifies the candlestick patterns and visualises them to the user. The secondary aim was also met. Although it was not to the desired level, the results showed that there are explanation pages with explanations, examples and images for each candlestick pattern used in the application.

I have achieved an application that effectively provides veteran traders with a supplementary tool for their financial analysis. It allows them to identify all instances of candlestick patterns on the chart and even multiple candlestick patterns at the same time. I have also achieved an application that invites new traders to learn how to use candlestick patterns in a beginner-friendly manner.

Reflection on Learning

Throughout this project, there were many things I learned about both the topic of the project and the difficulties of the process. This section goes through a few of the things I've learned.

Difficulty of Identifying Candlesticks

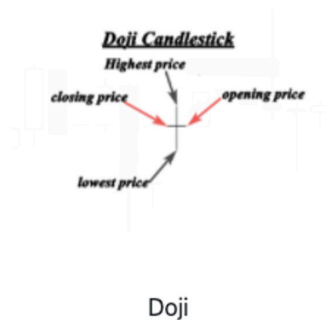


Figure 66: Reflection on Candlestick Patterns

The figure above shows the Doji candlestick pattern. This occurs when the closing and opening prices are the same. In theory, it's simple but when attempting to calculate it from data it becomes far more difficult. That is because there are almost no examples where the opening and close prices are the same. They are always close but not the same. This is especially the case when the prices in the data go to 5 or 6 decimal places. If I round to 1 decimal place then for currency pairs such as SHIB/USD, every candlestick will be Doji and for currency pairs with much higher prices such as BTC/USD, there would be no Doji candlesticks. Calculating candlestick patterns relative to the scale was the most difficult part. This made me realise that software such as machine learning pattern recognition may be more suitable to identify candlestick patterns instead which is in contrast to my original assumption.

Time Management

For a project spanning 12 weeks, time management is essential in completing all of the work by the final deadline. Throughout this project, I have learnt that it is extremely difficult and optimistic to try and predict how long a piece of functionality will take to code if you have not done it before. There are some functionalities such as the search bar that took far less time than expected to implement. However, the greatest discovery was that the bugs, errors, and slight improvements to the functionalities were often what consumed the most time. Had I not given myself some slack time in my initial plan to catch up then I would likely not have completed this project in time.

Supervisor Meetings

The supervisor meetings throughout the project became milestones for me to follow and ensured that a certain amount of work was completed between each meeting. This helped me a lot in maintaining steady progress. This has in turn made me realise the value of agile methods such as scrum meetings where progress is being tracked constantly and small milestones and mini-deadlines are scattered throughout the project.

Agile Methodology

For any projects I complete in the future, whether personal or commercial, I will likely be using agile methods. This is because I believe that the iterative development of features made the features a lot more complete and at a higher level. It also gave me milestones to

track my progress. Additionally, this made the code more organised. Most importantly, it kept me focused on implementing features based on the design to meet the criteria of the use cases and requirements.

Difficulties in motivation

Throughout all my years in education, I have never had such a large assignment. The difficulty in staying consistent and doing a little every day gradually became harder and harder. I did take breaks at times, but this past semester has been the most consistent I have ever been and that discipline of doing a little a day is a skill that I have tempered well throughout the course of this project. There were plenty of times such as the prototyping and implementation when I realised that my previous assumptions were wrong. I allocated a lot of time to tasks that turned out to be simple and less time to tasks that were much more difficult than anticipated. The latter often caused me anxiety. I began to lose trust in my plans while becoming worried about whether I would complete the project in time. However, I often found the best solution to this was to keep going and push through. This project has tested my perseverance time and time again and it has now become a skill I am quite proud of. Truly, motivation is temporary. Discipline and perseverance achieve results.

Works Cited

- Atlassian, n.d. *What is Version Control?*. [Online]
Available at: <https://www.atlassian.com/git/tutorials/what-is-version-control>
- Bank Of England, 2021. *How has Covid affected global trade?*. [Online]
Available at: <https://www.bankofengland.co.uk/bank-overground/2021/how-has-covid-affected-global-trade#:~:text=The%20Covid%20pandemic%20has%20caused,since%20the%20global%20financial%20crisis.>
- CandleScanner, n.d. *CandleScanner*. [Online]
Available at: <https://www.candlescanner.com/>
- Croy, M. H., 2020. *How To Calculate Time Complexity With Big O Notation*. [Online]
Available at: <https://medium.com/dataseries/how-to-calculate-time-complexity-with-big-o-notation-9afe33aa4c46>
- Fernando, J., 2021. *Law of Supply and Demand*. [Online]
Available at: <https://www.investopedia.com/terms/l/law-of-supply-demand.asp#:~:text=The%20law%20of%20demand%20says,are%20traded%20on%20a%20market.>
- Hayes, A., 2022. *Financial Markets*. [Online]
Available at: <https://www.investopedia.com/terms/f/financial-market.asp>
- Jabotinsky, H. Y. & Sarel, R., 2021. *How the Covid-19 Pandemic Affected the Cryptocurrency Market*. [Online]
Available at: <https://clsbluesky.law.columbia.edu/2021/03/26/how-the-covid-19-pandemic-affected-the-cryptocurrency-market/>
- Kaspersky, n.d. *What is cryptocurrency and how does it work?*. [Online]
Available at: <https://www.kaspersky.com/resource-center/definitions/what-is-cryptocurrency>
- Kenton, W., 2021. *Security*. [Online]
Available at:

<https://www.investopedia.com/terms/s/security.asp#:~:text=Securities%20are%20fungible%20and%20tradable,aspects%20of%20debt%20and%20equity.>

Murphy, C., 2021. *Support and Resistance Basics*. [Online]

Available at: <https://www.investopedia.com/trading/support-and-resistance-basics/#toc-support-and-resistance-defined>

Nielsen, J., 1994. *10 Usability Heuristics for User Interface Design*. [Online]

Available at: <https://www.nngroup.com/articles/ten-usability-heuristics/>

niko_83, 2020. *Candlestick Pattern Finder*. [Online]

Available at: <https://www.tradingview.com/script/xsPWmkt-Candlestick-Pattern-Finder/>

Pines, L., 2022. *Doji Formations: Learn How to Interpret Them to Help Trading Strategies*. [Online]

Available at: <https://commodity.com/technical-analysis/doji/>

Plotly, n.d. *Candlestick Charts in Python*. [Online]

Available at: <https://plotly.com/python/candlestick-charts/>

Polianskaya, A., 2018. *Humans may have been trading with each other as far back as 300,000 years*. [Online]

Available at: <https://inews.co.uk/news/science/early-humans-trading-300000-years-135655>

Raut, S., 2020. *The Candle Stick Trading Bible*. s.l.:s.n.

Reinicke, C., 2021. *1 in 10 people currently invest in cryptocurrencies, many for ease of trading, CNBC survey finds*. [Online]

Available at: <https://www.cnn.com/2021/08/24/1-in-10-people-invest-in-cryptocurrencies-many-for-ease-of-trading.html>

Singh, N., n.d. *How Many Stock Traders Are There?*. [Online]

Available at: <https://tradeveda.com/how-many-stock-traders-are-there/>

Smart Learning Studio, 2020. *Software Development Methodologies*. [Online]

Available at: <https://youtu.be/0mUyAWXwcYQ>

Start Bootstrap, n.d. *Blog Home*. [Online]

Available at: <https://startbootstrap.com/template/blog-home>

The Data Visualisation Catalogue, n.d. *Candlestick Chart*.

s.l.:https://datavizcatalogue.com/methods/candlestick_chart.html.

TwelveData, 2020. *twelvedata-python*. [Online]

Available at: <https://github.com/twelvedata/twelvedata-python>