Financial Investment Assistant

Dissertation

Tharidhu Angodage

[tmpa1@student.le.ac.uk](mailto:tmpa1@student.le.ac.uk)



School of Computing and Mathematical Sciences

CO3201 Computer Science Project

May 3, 2024

## Abstract

Calzone Capital, a Financial Investment Assistant designed to demystify the complexities of the stock market for novice investors. The platform integrates real-time stock data, current market news, and advanced predictive analytics to provide a user friendly interface for informed decision making. At its core, Calzone Capital employs a sophisticated LSTM model, which allows it to offer not just real time insights but also predictive analytics on stock price movements, enhancing user engagement and understanding of investment strategies.

Calzone Capital aims to bridge the gap between complex financial information and beginner investors by offering a streamlined, accessible gateway into the world of investing. The platform’s intuitive design is built towards simplifying the investment process, ensuring that users can easily navigate through different features such as interactive dashboards, comprehensive market summaries, and detailed stock analyses. This approach not only aids in making informed investment decisions but also cultivates a learning environment where users can gradually understand and react to market dynamics.

The development of this platform represents a fusion of innovative technology and user-centered design principles. Throughout the dissertation, the challenges encountered during the development process are explored, alongside the strategic decisions made to overcome these challenges and meet the project goals. The integration of real-time data, coupled with the predictive capabilities of the LSTM model, provides users with actionable insights, making Calzone Capital a valuable tool for emerging market participants.

## Declaration

All sentences or passages quoted in this report, or computer code of any form whatsoever used and/or submitted at any stages, which are taken from other people’s work have been specifically acknowledged by clear citation of the source, specifying author, work, date and page(s). Any part of my own written work, or software coding, which is substantially based upon other people’s work, is duly accompanied by clear citation of the source, specifying author, work, date and page(s). I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this module and the degree examination as a whole.

Name: Tharidhu Angodage

Date: 02.01.2024

Table of Contents

[abstract 2](#_Toc165636714)

[declaration 2](#_Toc165636715)

[1. introduction 4](#_Toc165636716)

[1.1. relevance and motivation 4](#_Toc165636717)

[1.2. aims and objectives 4](#_Toc165636718)

[1.3. challenges 4](#_Toc165636719)

[1.4. reading plan 5](#_Toc165636720)

[2. survey of literature 5](#_Toc165636721)

[2.1. logical background 5](#_Toc165636722)

[2.2. technical background 8](#_Toc165636723)

[3. requirements and specifications 8](#_Toc165636724)

[3.1. functional requirements 8](#_Toc165636725)

[3.2. non-functional requirements 9](#_Toc165636726)

[4. software architecture and design 10](#_Toc165636727)

[4.1 software architecture 10](#_Toc165636728)

[4.2 design 17](#_Toc165636729)

[5. algorithmic insights 20](#_Toc165636730)

[5.1 data acquisition and preprocessing 20](#_Toc165636731)

[5.2 lstm model architecture and training 21](#_Toc165636732)

[5.3 predictive model outputs 21](#_Toc165636733)

[5.4 recommendation logic 21](#_Toc165636734)

[5.5 investment strategies logic 22](#_Toc165636735)

[6. testing and evaluation 23](#_Toc165636736)

[6.1 testing 23](#_Toc165636737)

[6.2 evaluation of testing strategies 27](#_Toc165636738)

[6.3 challenges and limitations 27](#_Toc165636739)

[6.4 future directions 28](#_Toc165636740)

[7. critical appraisal 28](#_Toc165636741)

[7.1 project execution analysis 28](#_Toc165636742)

[7.2 achievements and shortcomings 29](#_Toc165636743)

[7.3 future directions 29](#_Toc165636744)

[7.4 discussion of social, sustainability, commercial, and economic context 30](#_Toc165636745)

[7.5 personal development assessment 30](#_Toc165636746)

[8. conclusion 31](#_Toc165636747)

[references 32](#_Toc165636748)

## 

## 1. Introduction

### 1.1. Relevance and Motivation

In today’s dynamic financial landscape, beginners often find it challenging to navigate the complexities of investment decisions. This Financial Investment Assistant website is designed with the novice investor in mind, offering a straightforward, user-friendly platform that integrates essential tools for real-time stock data, market news, and predictive analytics. By simplifying access to financial information through a clear and accessible interface, this tool makes the stock market easier to understand for users with limited financial knowledge.

Incorporating a predictive analytics feature based on a Long Short-Term Memory (LSTM) model, the website aligns with current trends in artificial intelligence to support easy and informed investment decisions. Additionally, the site offers a graphical comparison of different investment strategies, demonstrating potential returns to help users make educated choices about their finances. This project is specifically crafted to empower new investors, providing them with the tools to confidently manage their investments and understand market dynamics, making it an essential tool for today's emerging market participants.

### 1.2. Aims and Objectives

The primary aim of this project is to provide a beginner-friendly platform that simplifies the investment process by offering real-time financial data and predictive tools. The website is designed to assist novice investors in making informed decisions quickly and effectively. The specific objectives are:

1. **Real-Time Stock Data**: Implement live updates of stock prices on the homepage every second and refresh data on the market page every minute to provide timely information to users.
2. **Up-to-Date Market News**: Continuously pull and display the latest market news as soon as it becomes available to keep users informed about current market trends.
3. **Company News Updates**: Provide the latest news related to companies and stocks searched on the market page, ensuring users have access to fresh and relevant information.
4. **Historical Stock Data Visualization**: Enable users to view and interact with historical stock price graphs, adjustable to display data for today, the past week, month, six months, year, or all available data.
5. **Predictive Stock Price Model**: Utilise a LSTM model to analyse historical data and predict future stock prices, providing recommendations (Buy, Hold, or Sell) based on percentage changes, Piotroski score and bollinger bands.
6. **Investment Strategy Comparison**: Use predictive analytics to demonstrate potential profits or losses under different investment strategies, helping users to choose the best strategy based on their risk tolerance and investment goals.
7. **Expert Stock Analysis**: Integrate insights from professional stock analysts, offering advice on whether to buy, sell, or hold stocks, featured on the homepage for trending stocks.

These objectives are designed to empower users with real-time data and predictive analytics, enhancing their ability to navigate the stock market with confidence and make strategic investment decisions.

### 1.3. Challenges

During the development of the Calzone Capital, several technical challenges were encountered. Firstly, integrating TensorFlow into the project proved difficult, particularly with the introduction of new Apple Silicon chips. The transition to using the 'TensorFlow-macos' library required significant adjustments, prompting a shift from a simple pip environment to a conda one. Secondly, the evolving nature of NextJS posed challenges, as the framework underwent updates, certain lines of code had to be modified or removed, so had frequent adaptations in the project's codebase. Lastly, connectivity issues arose when working within university environments such as the library or Percy Gee labs. Establishing a connection to the MongoDB database proved challenging, particularly when working within university environments such as the library or Percy Gee labs. Despite efforts, the connection could not be established successfully using the Eduroam network, which significantly impeded the testing and development process. Excluding the technical challenges encountered during the development of Calzone Capital, there were also several weeks where I faced health issues, requiring hospital admission and necessitating a month of rest. These periods of illness impacted the project timeline and required adjustments to accommodate recovery time.

### 1.4. Reading Plan

The first chapter of this work introduces Calzone Capital, setting out the project's objectives, significance, and scope. It explains the purpose of the platform and highlights the need for such a tool in today's financial landscape.

In the second chapter, the focus shifts to a thorough review of literature and existing platforms. This survey lays a theoretical and practical foundation, covering fundamental stock trading concepts, advanced predictive modelling with LSTM, and insights from established financial platforms like Etoro and Trading 212.

The third chapter details the system design and architecture of Calzone Capital. It describes the technical stack and design choices made, focusing on user interface design, database management, and the integration of various APIs and services.

Chapter four discusses the implementation phase, where the platform's coding, features, and functionalities are elaborated upon. This includes the integration of real-time data feeds, predictive analytics, and mechanisms for user interaction.

Testing and evaluation are covered in the fifth chapter, which explains the different testing methods used to validate the reliability and performance of the platform. This includes tests for frontend, middleware, and backend functionalities, along with the incorporation of user feedback for system improvement.

The sixth chapter explores the project's broader impact, discussing its potential effects on users and the market. It considers the social, sustainability, commercial, and academic implications of Calzone Capital.

The final chapter concludes the project, reflecting on the achievements and limitations encountered during development. It also suggests areas for future improvement and potential research directions in the field of financial technology, paving the way for further advancements.

## 2. Survey of Literature

### 2.1. Logical Background

In the logical background, we focus on understanding fundamental concepts and principles that form the backbone of stock trading. Here, we explore essential terms and concepts. By gaining insights into these fundamental elements, we lay the groundwork for a comprehensive understanding of financial markets. Investopedia's guide [1] provided valuable insights on stock trading basic terminology, trading strategies, and market analysis techniques.

#### 2.1.1 Fundamentals of Stock Trading

It is imperative to lay a solid foundation by understanding the fundamentals of stock trading. This involved familiarising with essential concepts that form the backbone of financial markets. In the world of stock trading, certain fundamental terms and concepts serve as building blocks for investors. These include:

* **Stock Exchange:** A controlled market where investors transact in financial instruments like stocks and bonds, allowing businesses to raise funds and investors to purchase and sell shares.
* **Stock Symbol:** Also known as a ticker symbol, it's a distinctive character combination used to distinguish the stock of a publicly traded company on a stock exchange.
* **Opening Price:** The price at which a stock trades for the first time at the start of a trading session, representing the initial trade when the market opens.
* **Closing Price:** The cost at which a stock is traded for the final time during a trading session, representing the final deal before the market closes.
* **High Price:** The highest price at which a stock trades during a specific trading session, indicating the stock's peak price within that period.
* **Low Price:** The lowest price at which a stock is traded during a specific trading session, indicating the stock's lowest price point.
* **Volume:** The total number of shares exchanged during a given time frame, such as a trading day or session.
* **Previous Close:** The price at which a stock traded at the conclusion of the previous trading session is referred to as the stock's previous close.

#### 2.1.2 Understanding LSTM Models

In the realm of predictive analytics, Long Short-Term Memory (LSTM) models play a crucial role in forecasting stock prices and making informed investment decisions. LSTM is a type of recurrent neural network (RNN) architecture that is specifically designed to capture long-term dependencies and patterns in sequential data [2]. Unlike traditional feedforward neural networks, LSTM models can retain information over extended time periods, making them well suited for time series prediction tasks such as stock price forecasting [3].

LSTM models consist of interconnected cells that process sequential input data over multiple time steps. Each cell contains three gates - the input gate, forget gate, and output gate, which regulate the flow of information through the network. These gates enable LSTM models to selectively retain or discard information based on its relevance to the prediction task. By learning from historical price data and incorporating external factors such as market trends and news sentiment, LSTM models can generate accurate predictions of future stock prices.

In the context of stock trading, LSTM models are trained on historical price data to identify patterns and trends that can inform future price movements. By analysing sequential price data and extracting relevant features, LSTM models can predict potential price changes and provide valuable insights for traders and investors. Additionally, LSTM models can be combined with other technical indicators and fundamental analysis techniques to enhance prediction accuracy and robustness.

#### 2.1.3. Generating Recommendations

In the process of generating recommendations, several key metrics and indicators are utilised to assess the financial health and market dynamics of a stock. These include the Piotroski score, Bollinger Bands, and percentage change.

**Piotroski Score**

The Piotroski score is a financial metric developed by Joseph Piotroski, designed to assess the strength of a company's financial position [4]. It evaluates various aspects of a company's financial statements, such as profitability, leverage, operating efficiency, and liquidity. A higher Piotroski score indicates stronger financial health, while a lower score may signify potential weaknesses.A black text on a white background

Description automatically generated

The following abbreviations were used in the formula and calculation of Piotroski F-Score:

* ROA - Return on Assets
* CFO - Operating Cash Flow
* LTD - Long Term Debt
* CY - Current Year
* PY - Previous Year
* ATR - Asset Turnover Ratio

**Bollinger Bands**

A diagram of mathematical equations

Description automatically generatedA diagram of a stock market

Description automatically generatedBollinger Bands are a technical analysis tool used to measure a stock's volatility and potential price movements [5]. They consist of three lines: the middle line, which represents the simple moving average (SMA) of the stock's price; the upper band, which is calculated by adding a multiple of the standard deviation of the price to the SMA; and the lower band, calculated by subtracting the same multiple of the standard deviation from the SMA. Bollinger Bands help identify overbought and oversold conditions in the market, with prices above the upper band suggesting overbought conditions and prices below the lower band indicating oversold conditions.

**Percentage Change**

Percentage change is a simple yet powerful metric used to track the relative change in a stock's price over a specific period. It is calculated by dividing the difference between the current price and the previous price by the previous price, then multiplying by 100 to express the change as a percentage. Positive percentage changes indicate price appreciation, while negative changes signify price depreciation. Analysing percentage changes allows traders to gauge the momentum and direction of price movements, helping inform trading decisions [6].

### 2.2. Technical Background

The project began with a thorough examination of established platforms like Etoro [7] and Trading 212. This detailed review aimed to uncover the features and functionalities provided by these platforms, laying the groundwork for a comprehensive understanding of financial concepts.

Recognizing the significance of programming languages and libraries in finance, resources such as "Python for Finance" [8] were instrumental in selecting Python as the programming language for the platform. Further honing programming skills and understanding the application of machine learning in trading were achieved through the Udacity course on "Machine Learning for Trading" [9]. This course provided valuable insights into trading strategies and the utilisation of Python libraries for financial analysis.

Despite the preparatory steps taken, delving into the realm of web development posed its challenges, particularly in mastering NextJS. With limited prior experience in this framework, YouTube courses as "Full Modern React Tutorial” by Academind [10] were necessary to gain proficiency in NextJS. Moreover, navigating the intricacies of testing within the NextJS environment was unfamiliar territory, requiring dedicated effort to acquire this skill set.

Another significant endeavour was the adoption of Long Short-Term Memory (LSTM) models for predictive analytics. Although initially intimidating, exploring the intricacies of LSTM models provided a profound comprehension of their significance in predicting stock prices and making informed investment choices with foundational concepts reinforced by "Deep Learning for Time Series Forecasting” [11]. Furthermore, the process of designing and implementing custom UI designs into NextJS using Tailwind CSS demanded meticulous attention to detail and a creative approach. Resources such as "Tailwind CSS: From Zero to Production" [12] were invaluable in overcoming this challenge required significant time investment to seamlessly integrate the UI designs with the functionality of the platform.

In summary, the technical background of Calzone Capital's development journey involved overcoming various hurdles, from mastering NextJS and LSTM models to crafting UI designs. Each challenge was met with determination and a commitment to learning, ultimately contributing to the project's growth and evolution.

## 3. Requirements and Specifications

This section outlines the technical and functional requirements necessary to ensure the Financial Investment Assistant website operates efficiently and meets user expectations for functionality and performance. Each feature is built to provide users with real-time, actionable financial data and insights.

### 3.1. Functional Requirements

1. User Authentication
   * **Requirement** - Secure user authentication system that supports login and registration functionality.
   * **Specification** - Utilise NextAuth.js for handling credential based authentication. All sensitive user data, including credentials, are securely stored in MongoDB, with password hashing and session management handled efficiently to ensure security and integrity.
2. Real-Time Stock Data
   * **Requirement** - Automatic updating of stock prices on the homepage every second and on the market page every time the page is refreshed, with data reflecting the latest available updates at least every minute.
   * **Specification** - Implement WebSocket to enable a continuous and real-time data stream from *twelvedata API*, ensuring that stock prices on the homepage are updated every second. In the market page, utilise standard API calls (from *twelvedata API*) to pull the latest stock data when the market page is accessed or manually refreshed by the user.
3. Real-Time Market News
   * **Requirement** - Continuously update the news section with the latest financial market news.
   * **Specification** - Integrate the *Finnhub API* to pull the latest five market news articles each time the homepage is refreshed.
4. Company News Updates
   * **Requirement** - Fetch and display the latest news articles related to specific companies or stocks on demand.
   * **Specification** - Use the *Finnhub API* to fetch the latest four news articles related to the company. The news is retrieved based on a one week interval from the current date, ensuring the relevance and timeliness of the information provided.
5. Historical Stock Graph
   * **Requirement** - Allow users to view and manipulate historical stock price graphs over various time ranges.
   * **Specification** - Utilise the Recharts to create interactive charts. Enabling users to dynamically view historical stock data across different time frames such as today, 1 week, 1 month, 6 months, 1 year, and all-time.
6. Stock Prediction and Recommendation
   * **Requirement** - Provide future stock price predictions and recommendations (Buy, Hold, Sell) for the selected stock.
   * **Specification** - Develop and integrate an LSTM model to forecast future stock prices based on historical data. The recommendation system leverages three key indicators: Percentage Change, Piotroski Score and Bollinger Bands. For instance, a significant increase in price might trigger a 'Sell' recommendation, whereas a significant decrease might suggest a 'Buy'. These predictive insights are combined to provide users with informed and actionable investment guidance.
7. Investment Strategies
   * **Requirement** - Calculate and visually compare returns for different investment strategies based on model predictions. Users can enter a total investment amount, which the system allocates over selected periods
   * **Specification** - Use the LSTM model’s predictions to enable users to compare weekly and monthly investment strategies. Calculate the potential returns and visually displays the effectiveness of each strategy relative to the initial investment, helping users make informed decisions about their investment approach.

### 3.2. Non-Functional Requirements

1. Performance:
   * **Requirement**: The website should load and respond quickly, even during peak usage.
   * **Specification**: Optimise backend services and use a content delivery network (CDN) to enhance load times and data retrieval speeds.
2. Usability:
   * **Requirement**: The website must be easy to navigate and intuitive, especially for users with limited investment experience.
   * **Specification**: Implement a clean, user-friendly UI design, ensuring the website is accessible and responsive on various devices.
3. Security:
   * **Requirement**: High-level security for user data, especially financial information.
   * **Specification**: Implement HTTPS, password hashing, and secure storage of sensitive data in MongoDB.
4. Scalability:
   * **Requirement**: The system must handle an increasing number of users and data volume without degradation in performance.
   * **Specification**: Design the architecture to be scalable, using cloud services and load balancers.

## 4. Software Architecture and Design

This chapter offers an in depth exploration of the project's software architecture and system design intricacies. It traces the workflow of the system from start to finish. Through this exploration, the chapter aims to provide a detailed understanding of the project's technological framework and design choices, essential for creating this Financial Investment Assistant.

### 4.1 Software Architecture

The Software Architecture provides a fundamental overview of the technical framework supporting the project. Here, we delve into the specific technological choices made to construct a robust and adaptable system. As shown in Fig 4.1, it lays the basis for a thorough examination of how numerous technologies, libraries, frameworks, and databases interact. A diagram of a company

Description automatically generated

The implemented system comprises three primary components: Frontend, Backend, and Database.

In the **Frontend** user interface, we've utilised JavaScript technologies, particularly Next.js, to deliver a seamless and interactive user experience. Through integration with APIs from TwelveData [13] and Finnhub [14], users can access crucial financial stock information, including historical price data and company related news.

**Backend** functionality is pivotal to the application, serving as its core component. It comprises two key elements: Business Logic and Machine Learning Algorithms, each playing a distinct role in the system:

* **Business Logic:**
* Implemented in Next.js, the business logic serves as a crucial intermediary between the Frontend and Data storage. It incorporates endpoints and APIs responsible for gathering data, performing necessary calculations, and serving the processed data to the Frontend for user display.
* **Machine Learning Algorithms:**
  + These algorithms, developed in Python (specifically Flask), are dedicated to analysing stock data obtained from the *twelvedata API*. They return predicted prices, including historical data, and offer recommendations and strategies tailored around the LSTM model to guide investment decisions effectively. Python's extensive range of machine learning modules facilitates quick and accurate solutions to the complex formulations and large datasets inherent in these algorithms.

Together, Next.js and Python enable seamless data processing and interaction between the backend and frontend, enhancing the overall functionality of the platform.

The **Database**, built on MongoDB, serves as the backbone of our system architecture. Its primary function is to securely store and organise user credentials, ensuring efficient access and retrieval when needed.

#### 4.1.1 User Interface

The user interface is crafted using Next.js, which enables efficient server-side rendering and static generation. Unlike standard approaches, I custom-built all UI components directly with Next.js, avoiding third-party UI libraries to ensure a unique and streamlined user experience. For styling, Tailwind CSS was utilised. This method provides extensive control over the aesthetics and functionality, ensuring that each page aligns with specific performance and user engagement goals. Overall, the UI is designed to be intuitive and responsive, facilitating seamless real-time interactions with financial data without overwhelming amateur users.

#### 4.1.2 Middleware Services

The middleware services are developed using Next.js. This section of the project encapsulates crucial aspects of the system including business logic, stock data retrieval via APIs, and real-time data streaming through a WebSocket. This architecture ensures efficient data handling and provides a dynamic user experience by delivering live data directly to the frontend without requiring page refreshes.

* Real-Time Data via Websocket

const socket = new WebSocket(`wss://[ws.twelvedata.com/v1/quotes/price?apikey=API\_KEY](http://ws.twelvedata.com/v1/quotes/price?apikey=API_KEY)`);

   socket.addEventListener('open', (event) => {

     socket.send(JSON.stringify({

       action: 'subscribe',

       params: { symbols: symbols.join(',') }

     }));

   });

   socket.addEventListener('message', (event) => {

     const data = JSON.parse(event.data);

     onMessage(data);

   });

The code above establishes a WebSocket connection to the *TwelveData* service at the provided URL, which enables real-time streaming of stock price quotes. The connection uses the WebSocket protocol (wss://) to ensure secure, encrypted communication, and the provided API key authenticates access to the service. Once connected, the WebSocket can receive live updates for stock prices, facilitating real-time data display on your website.

In the following sections, we will explore the various API pulls utilised within the platform. The API interactions are primarily conducted through two main data sources: *Finnhub* and *TwelveData*. For clarity, throughout our documentation, basePath refers to the API endpoint for *Finnhub*, while twelveDataBasePath is designated for the *TwelveData* API endpoints. These APIs are integral to fetching real-time and historical financial data, enabling our application to deliver timely and accurate market information to users.

* GET /api/search/{q}
  + Endpoint to search for stock symbols**.** This API endpoint facilitates the querying of stock symbols based on user input. The query is specified via the URL parameter ‘q’.

Sample request and response for the AAPL ticker:

const url = `${basePath}/search?q=AAPL&token=API\_KEY`;

{

 "count": 4,

 "result": [

   {

     "description": "APPLE INC",

     "displaySymbol": "AAPL",

     "symbol": "AAPL",

     "type": "Common Stock"

   },

   {

     "description": "APPLE INC",

     "displaySymbol": "AAPL.SW",

     "symbol": "AAPL.SW",

     "type": "Common Stock"

   },

   {

     "description": "APPLE INC",

     "displaySymbol": "APC.BE",

     "symbol": "APC.BE",

     "type": "Common Stock"

   },

   {

     "description": "APPLE INC",

     "displaySymbol": "APC.DE",

     "symbol": "APC.DE",

     "type": "Common Stock"

   }

 ]

}

* GET /api/stock/profile2/{symbol}
  + Endpoint to retrieve detailed stock profiles. This API endpoint allows for the fetching of comprehensive stock details based on the stock symbol. The stock symbol is specified via the URL parameter ‘symbol’.

Sample request and response for the AAPL ticker:

const url = `${basePath}/stock/profile2?symbol=AAPL&token=API\_KEY`;

{

 "country": "US",

 "currency": "USD",

 "exchange": "NASDAQ/NMS (GLOBAL MARKET)",

 "ipo": "1980-12-12",

 "marketCapitalization": 1415993,

 "name": "Apple Inc",

 "phone": "14089961010",

 "shareOutstanding": 4375.47998046875,

 "ticker": "AAPL",

 "weburl": "https://www.apple.com/",

 "logo": "https://static.finnhub.io/logo/87cb30d8-80df-11ea-8951-00000000092a.png",

 "finnhubIndustry":"Technology"

}

* GET /api/quote/{symbol}
  + Endpoint to retrieve stock quotes. This API allows for the fetching of real-time stock quotes based on the stock symbol. The stock symbol is specified via the URL parameter ‘symbol’.

Sample request and response for the AAPL ticker:

const url = `${basePath}/quote?symbol=AAPL&token=API\_KEY`;

{

  "c": 261.74,

  "h": 263.31,

  "l": 260.68,

  "o": 261.07,

  "pc": 259.45,

  "t": 1582641000

}

* GET /api/time\_series/{symbol}/{interval}/{outputSize}
  + Endpoint to retrieve stock time series. This API endpoint provides time series data and metadata for a specified stock symbol. The stock symbol and the time interval for the data are specified through the ‘symbol’ and ‘interval’ URL parameters, respectively. Additionally, the ‘outputSize’ parameter allows users to define the number of data points returned. The response includes a ‘meta’ object with general information about the stock and a ‘time series’ array that contains detailed price data (Open, High, Low, Close) ordered by time in descending sequence.

Sample request and response for the AAPL ticker:

const url = `${twelveDataBasePath}/time\_series?symbol=AAPL&interval=1min&apikey=API\_KEY&outputsize=2`;

{

  "meta": {

    "symbol": "AAPL",

    "interval": "1min",

    "currency": "USD",

    "exchange\_timezone": "America/New\_York",

    "exchange": "NASDAQ",

    "mic\_code": "XNAS",

    "type": "Common Stock"

  },

  "values": [

    {

      "datetime": "2021-09-16 15:59:00",

      "open": "148.73500",

      "high": "148.86000",

      "low": "148.73000",

      "close": "148.85001",

      "volume": "624277"

    },

    {

      "datetime": "2021-09-16 15:58:00",

      "open": "148.72000",

      "high": "148.78000",

      "low": "148.70000",

      "close": "148.74001",

      "volume": "274622"

    }

  ],

  "status": "ok"

}

* GET /api/news/{category}
  + Endpoint to retrieve stock market news. This API call is designed to fetch the latest stock market news based on the specified category. On our platform, using the parameter ‘general’ allows users to access the most recent general market news.

Sample request and response for the ‘general’ market news:

const url = `${basePath}/news?category=general&token=API\_KEY`;

 {

    "category": "technology",

    "datetime": 1596589501,

    "headline": "Square surges after reporting 64% jump in revenue, more customers using Cash App",

    "id": 5085164,

    "image": "https://image.cnbcfm.com/api/v1/image/105569283-1542050972462rts25mct.jpg?v=1542051069",

    "related": "",

    "source": "CNBC",

    "summary": "Shares of Square soared on Tuesday evening after posting better-than-expected quarterly results and strong growth in its consumer payments app.",

    "url": "https://www.cnbc.com/2020/08/04/square-sq-earnings-q2-2020.html"

  },

  {

    "category": "business",

    "datetime": 1596588232,

    "headline": "B&G Foods CEO expects pantry demand to hold up post-pandemic",

    "id": 5085113,

    "image": "https://image.cnbcfm.com/api/v1/image/106629991-1595532157669-gettyimages-1221952946-362857076\_1-5.jpeg?v=1595532242",

    "related": "",

    "source": "CNBC",

    "summary": "\"I think post-Covid, people will be working more at home, which means people will be eating more breakfast\" and other meals at home, B&G CEO Ken Romanzi said.",

    "url": "https://www.cnbc.com/2020/08/04/bg-foods-ceo-expects-pantry-demand-to-hold-up-post-pandemic.html"

  },

* GET /api/company-news/{symbol}/{from}/{to}
  + Endpoint to retrieve specific company news. This API call is designed to fetch news articles related to a specified stock symbol, filtered between two dates. The dates are dynamically formatted to retrieve news for a specific period between the current date and seven days prior.

Sample request and response for the AAPL ticker:

const url = `${basePath}/company-news?symbol=AAPL&from=2023-08-15&to=2023-08-20&token=API\_KEY`;

[

  {

    "category": "company news",

    "datetime": 1569528720,

    "headline": "How to disable comments on your YouTube videos in 2 different ways",

    "id": 25287,

    "image": "https://amp.businessinsider.com/images/5d8d16182e22af6ab66c09e9-1536-768.jpg",

    "related": "AAPL",

    "source": "Business Insider",

    "summary": "You can disable comments on your own YouTube video if you don't want people to comment on it. It's easy to disable comments on YouTube by adjusting the settings for one of your videos in the beta or classic version of YouTube Studio. Visit Business Insider's homepage for more stories . The comments section has a somewhat complicated reputation for creators, especially for those making videos on YouTube . While it can be useful to get the unfiltered opinions of your YouTube viewers and possibly forge a closer connection with them, it can also open you up to quite a bit of negativity. So it makes sense that there may be times when you want to turn off the feature entirely. Just keep in mind that the action itself can spark conversation. If you decide that you don't want to let people leave comments on your YouTube video, here's how to turn off the feature, using either the classic or beta version of the creator studio: How to disable comments on YouTube in YouTube Studio (beta) 1. Go to youtube.com and log into your account, if necessary. 2.",

    "url": "https://www.businessinsider.com/how-to-disable-comments-on-youtube"

  },

  {

    "category": "company news",

    "datetime": 1569526180,

    "headline": "Apple iPhone 11 Pro Teardowns Look Encouraging for STMicro and Sony",

    "id": 25341,

    "image": "http://s.thestreet.com/files/tsc/v2008/photos/contrib/uploads/ba140938-d409-11e9-822b-fda891ce1fc1.png",

    "related": "AAPL",

    "source": "TheStreet",

    "summary": "STMicroelectronics and Sony each appear to be supplying four chips for Apple's latest flagship iPhones. Many other historical iPhone suppliers also make appearances in the latest teardowns….STM",

    "url": "https://realmoney.thestreet.com/investing/technology/iphone-11-pro-teardowns-look-encouraging-for-stmicro-sony-15105767"

  },

]

* GET /api/stock/recommendation/{symbol}
  + Endpoint to retrieve latest analyst recommendation trends**.** This API returns up-to-date analyst recommendation trends for a specific company, identified by its stock symbol.

Sample request and response for the AAPL ticker:

const url = `${basePath}/stock/recommendation?symbol=AAPL&token=API\_KEY`;

[

  {

    "buy": 24,

    "hold": 7,

    "period": "2020-03-01",

    "sell": 0,

    "strongBuy": 13,

    "strongSell": 0,

    "symbol": "AAPL"

  }

]

Now, let's delve into our Flask Python API, a key tool for the platform's functionality. This API retrieves stock predictions and investment strategies generated by our LSTM model, developed using various Python libraries. Moreover, it also accesses and retrieves user data stored in MongoDB.

* GET /predict/new-model/{symbol}/{timeframe}
  + Endpoint to retrieve stock predictions. This API endpoint allows users to request stock price predictions using the machine learning model. By sending the parameters ‘symbol’ and ‘timeframe’, users can obtain predictions for the specified stock symbol within the designated time frame.

const response = await fetch(`[http://127.0.0.1:5001/predict/new-model?symbol=${stockSymbol}&timeframe=${timeframe](http://127.0.0.1:5001/predict/new-model?symbol=$%7BstockSymbol%7D&timeframe=$%7Btimeframe)}`,

* GET /strategies/{symbol}/{timeframe}/{investment}
  + Endpoint to retrieve investment strategies. This API endpoint allows users to request investment strategies using the machine learning model. By sending the parameters ‘symbol’ and ‘timeframe’ and ‘investment’, users can obtain the gains or losses associated with different investment strategies for the specified stock symbol within the designated timeframe and with the desired total investment amount.

const response = await fetch(`http://127.0.0.1:5001/strategies?symbol=${stockSymbol}&timeframe=${timeframe}&investment=${investmentAmount}`,

* GET /userInfo/{email}
  + Endpoint to fetch the user’s information. This API endpoint returns user details based on the provided email address, including their first name, last name, date of birth, phone number, stock wishlist, and account balance.

const res = await fetch(`http://localhost:5000/userInfo?email=${encodeURIComponent(userEmail)}`)

4.1.3 Database

The MongoDB system utilises the Mongoose library to structure and manage user data within the User collection efficiently. The schema comprises fields such as email, password, firstName, lastName, dob, phoneNumber, stockWishlist, and balance. Each user document is uniquely identified by email, and sensitive data like passwords are securely stored using bcrypt’s hashing function.

A screenshot of a computer

Description automatically generated

The schema also employs MongoDB's timestamps feature, which automatically manages createdAt and updatedAt fields, tracking each document's lifecycle. During the user registration process, the system connects to MongoDB, hashes the user’s password for enhanced security, and creates a new user document, ensuring secure data management.

   const { email, password, firstName, lastName, dob, phoneNumber } = await req.json();

   const hashedPassword = await bcrypt.hash(password, 10);

   await connect();

   await User.create({

     email,

     password: hashedPassword,

     firstName,

     lastName,

     dob,

     phoneNumber,

   });

For user authentication, the system employs NextAuth with a custom credentials provider. This setup involves retrieving the user document by email and comparing the stored hashed password with the provided one using bcrypt, thus ensuring secure and efficient user verification.

The MongoDB configuration is designed for robustness and scalability, supporting high performance operations essential for maintaining a responsive user experience. This architecture underpins the application, securing efficient and reliable data handling and management.

### 4.2 Design

User Interface Implementation

This section details the graphical user interface of the system. It includes descriptions and screenshots of each page, showing the layout, design elements, and functional features. This section aims to show how the interface supports user interaction and improves the user experience by combining functionality with visual design. Each screenshot is annotated to highlight important components and explain how they function within the application.

#### A close-up of a logo Description automatically generated4.2.1. Landing Page, Registration and Logging In

The landing page of Calzone Capital presents a sleek and inviting interface, characterised by a clean and modern aesthetic that immediately captures user attention. Prominent yet straightforward navigation buttons for 'Get Started' and 'Log In' offer clear pathways for user engagement.

A screen shot of a login form

Description automatically generatedThe registration form shown is designed for clear navigation and simplicity, leading users through an uncomplicated sign-up procedure. It includes standard fields such as email address, password, name, date of birth, and phone number, each designed to be intuitive for the user to complete. Advanced validations are in place to ensure the integrity of the data entered: these include verifying the uniqueness of the email address, checking that the password meets complexity requirements, confirming the user is at least 18 years old, and validating that the phone number contains the correct number of digits. Such measures are crucial for maintaining the security and reliability of user data from the outset.

A screen shot of a login form

Description automatically generatedThe login form embodies a minimalist and user friendly design. With a clear and focused layout, it facilitates a quick and effortless login experience. An alternative link to 'Register Here' is strategically placed for new users, ensuring a smooth navigation path for both returning and prospective users looking to explore the financial platform.

#### A screenshot of a computer Description automatically generated4.2.2. Home Page

The homepage of Calzone Capital is strategically organised to immerse users in the financial world with immediate effect. Central to this experience is the "Live Price" section, powered by WebSocket technology, offering users instant access to real-time financial data for a range of securities and currencies.

Adjacent to this, the "Market News" segment captures the latest, most relevant financial news stories, helping users stay informed with the most recent market developments. The careful curation of news articles supports informed decision-making by providing a constant stream of current financial insights.

Completing the homepage's offerings, the "Recommended Stocks" area presents analytics’ recommendations on trending stocks. This not only guides users on potential investment opportunities but also showcases Calzone Capital’s commitment to equipping its users with comprehensive analytical insights to inform their investment strategies.

#### 4.2.3. Sidebar

The sidebar serves as an efficient navigational tool, offering users straightforward access to key sections of the platform. It allows smooth transitions between the home page, market page and profile page. Additionally, the option to sign out is conveniently located within the sidebar, ensuring a secure and user-friendly experience as users navigate through the various features of Calzone Capital.

#### A screenshot of a computer Description automatically generated4.2.4. Market Page

The market page of Calzone Capital is a comprehensive hub where users can delve into the details of individual stocks. Central to this page is a search bar that allows users to look up specific stocks easily. The page features a responsive graph, which is a pivotal tool for users to visualise historical stock data. It provides the flexibility to view trends over various time frames, from a single day up to the entire history of available data.

At a glance, users can see an overview of the selected stock, including the current price, the change from the previous close in both numerical and percentage terms, which provides a snapshot of recent performance. Additionally, this page offers a list of the four latest news articles related to the stock, keeping investors informed with timely and relevant information.

Two prominent buttons invite users to explore predictions for future stock prices or to examine the potential of different investment strategies based on current trends. The layout and design of the market page are tailored for user-friendliness, ensuring that users of all levels can navigate and make use of the financial insights with ease.

#### 4.2.5. Stock Prediction & Strategies Evaluation

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedUpon selecting the 'Strategies' or 'Predict' buttons on the market page, users are met with a straightforward modal that enhances their investment planning experience. For 'Strategies', the modal prompts users to input a total investment amount and choose a timeframe. Similarly, when opting to 'Predict', users can select their desired forecast period, which facilitates a targeted insight into the future stock performance. The modals are designed with the user in mind, turning complicated financial choices into a few easy clicks for straightforward and accessible strategic planning.

A screenshot of a graph

Description automatically generatedAfter users have input their selections for stock prediction and strategy evaluation, a clear loading screen is displayed, indicating that the system is processing their request. During this brief wait, a gentle elevator music plays, which provides a soothing auditory backdrop. For users who prefer silence, there is a convenient mute button at the top right of the screen, offering control over the audio environment.

A screen shot of a graph

Description automatically generatedA screenshot of a graph

Description automatically generated  
The 'Strategies Results' modal graphically displays the projected performance of various investment strategies over time, allowing users to visually compare the outcomes of different approaches. This can assist users in making informed decisions tailored to their financial goals and risk preferences. The 'Prediction Results' modal offers a predictive graph illustrating potential future stock price movements. It presents both the current and predicted prices, including the percentage change, and provides a recommendation based on the analysis. This empowers users with forward-looking insights that are easy to interpret and act upon. Both modals are designed with simplicity in mind, making complex data and predictions accessible and comprehensible to users of all levels of financial expertise. The clear visual representations and straightforward language ensure that users find the information helpful and easy to understand, supporting smarter investment choices.

#### A screenshot of a profile Description automatically generated4.2.6. Profile Page

The profile page provides a straightforward display of user information, presenting essential details in a clean, organised layout. Here, users can view their first name, last name, email address, phone number, and date of birth. The design emphasises readability and simplicity, ensuring that personal data is easily accessible. Currently, the profile functions as a read only overview, without the option to edit within the platform.

## 

## 5. Algorithmic Insights

The application of machine learning in the stock market has revolutionised the way investors analyse and predict future stock prices. Long Short-Term Memory networks (LSTMs), a special kind of Recurrent Neural Network (RNN), are particularly adept at capturing time-series data's temporal dependencies. In this chapter, we explore the implementation and role of LSTMs in forecasting stock prices and the development of investment strategies within the Calzone Capital application.

### 5.1 Data Acquisition and Preprocessing

The financial data is sourced through the TwelveData API, a robust and comprehensive financial data service. The data retrieved includes daily closing prices and trading volumes, which are pivotal in predicting future stock performance. This data acquisition is the first critical step in our analytical process.Before feeding the raw data into our LSTM model, preprocessing is performed to ensure the quality and compatibility of the data. We utilise the ‘*MinMaxScaler’* from the ‘sklearn.preprocessing’ package to normalise the closing prices and volumes between 0 and 1. This normalisation is a standard practice that helps improve the convergence speed during training and the model's performance.

### 5.2 LSTM Model Architecture and Training

   x\_data, y\_data = np.array(x\_data), np.array(y\_data)

   splitting\_len = int(len(x\_data) \* 0.7)

   x\_train = x\_data[:splitting\_len]

   y\_train = y\_data[:splitting\_len]

   x\_test = x\_data[splitting\_len:]

   y\_test = y\_data[splitting\_len:]

   model = Sequential([

       LSTM(100, return\_sequences=True, input\_shape=(sequence\_length, 2)),

       Dropout(0.2),

       LSTM(50, return\_sequences=False),

       Dropout(0.2),

       Dense(1)

   ])

   model.compile(optimizer='adam', loss='mean\_squared\_error')

   model.fit(x\_train, y\_train, epochs=20, batch\_size=64, validation\_split=0.1)

The LSTM model is constructed using the Keras library. The Sequential model consists of an input layer followed by two LSTM layers with 100 and 50 neurons, respectively. Dropout layers with a rate of 0.2 are introduced to prevent overfitting. The output layer comprises a single neuron to predict the next day's closing price, using 'adam' as the optimizer and 'mean\_squared\_error' as the loss function.We train the model using historical stock price data, splitting it into training and test sets at a 70:30 ratio. The ‘fit’ method is used to train the model for 20 epochs with a batch size of 64, and a validation split of 0.1 to monitor the model's performance on unseen data during training.

### 5.3 Predictive Model Outputs

Upon the completion of training, the LSTM model accomplishes three key tasks: it reproduces the historical stock price data, predicts future stock prices based on the time frame selected, and provides a final prediction for the closing price at the end of the selected timeframe. These predictions are coupled with the corresponding dates to offer a comprehensive view of the stock's potential trajectory. The model's output includes the latest historical price and the predicted price, allowing for a direct comparison to assess the model's forecasting capability.

### 5.4 Recommendation Logic

The recommendation system in the application uses three key indicators to guide users in making stock decisions. It starts by looking at the Piotroski F-Score, a measure that rates a company’s financial health. If the score is high (above 6), it suggests that the company is financially strong, which might lead to a "Buy" recommendation. A low score (below 3) might mean financial trouble, suggesting a "Sell."

   # Check Piotroski score

   if piotroski\_score > 6:

       explanations.append(f"Piotroski score is strong at {piotroski\_score}.")

   elif piotroski\_score < 3:

       explanations.append(f"Piotroski score is weak at {piotroski\_score}.")

Next, the system checks the stock’s price movements against Bollinger Bands, which helps understand if the stock price is higher or lower than usual. If the stock price is above the upper Bollinger Band, the stock might be too expensive, and the system might recommend to "Sell." If it's below the lower band, it might be undervalued, and the system might suggest to "Buy."

   # Check Bollinger Bands

   if latest\_predicted\_price > upper\_band\_latest:

       explanations.append("Price is above the upper Bollinger Band, suggesting the stock may be overbought.")

       recommendation = "SELL"

   elif latest\_predicted\_price < lower\_band\_latest:

       explanations.append("Price is below the lower Bollinger Band, suggesting the stock may be oversold.")

       recommendation = "BUY"

The third factor is the percentage change in the stock price, which shows how much the price has gone up or down recently. A big increase might mean the stock is overbought, and it might be time to sell, while a big decrease might indicate it's a good time to buy.

   # Consider percentage changes last to allow override

   if percentage\_change > 5:

       explanations.append("Significant recent increase in stock price may indicate an overbought condition.")

       recommendation = "SELL"

   elif percentage\_change < -5:

       explanations.append("Significant recent decrease in stock price may indicate an oversold condition.")

       recommendation = "BUY"

The system combines these indicators to give a clear recommendation: "Buy," "Hold," or "Sell." This advice helps users make informed decisions by providing a balanced view based on the company’s financial state and recent stock price behaviour. This approach is designed to be straightforward, helping even users with limited investment experience to understand market trends and make smarter investment choices.

### 5.5 Investment Strategies Logic

The strategic investment algorithm within Calzone Capital is designed to provide users with a practical simulation of investment strategies, enabling them to assess potential returns over specified timeframes using stock price forecasts from the LSTM model. The key element of this feature is the flexibility it offers users in selecting their investment parameters and observing how different strategies could perform under varying market conditions.

The algorithm supports two primary investment strategies: Weekly and Monthly Investments. This simulation starts with the user specifying a total investment amount and a timeframe for the investment (3 months, 6 months, 1 year). These inputs are critical as they define the scope of the simulation and influence the strategy's potential outcomes. Upon receiving the user’s input for the total investment and desired timeframe, the system maps the timeframe to a corresponding number of days using a predefined mapping.

   # Convert the timeframe parameter to number of days

   time\_frame\_mapping = {

       "3 months": 60,

       "6 months": 120,

       "1 year": 240

   }

The total investment amount is then strategically divided across the selected timeframe. For Weekly Investments, the total is divided by the number of weeks within the timeframe, and similarly for Monthly Investments, it is divided by the number of months. This approach ensures that investments are distributed evenly over the period, allowing for a consistent application of funds which is crucial for assessing the compounded impact of the investments over time.

Using the function above, the algorithm simulates the investment process at each interval (weekly or monthly), calculating the number of shares that can be bought at the current predicted stock price. This method effectively shows how regular investments could accumulate and potentially grow, leveraging the power of compounding in a practical investment scenario. The strategy dynamically adjusts the shares owned as new investments are made at each interval, updating the investment's value based on the latest predictions from the LSTM model.

A screenshot of a graph

Description automatically generatedThe output displays a graphical view of this data in the frontend, with a straightforward chart that makes it easier to see the potential growth of the investments over time. This chart, with its neat and clear layout, lets users easily compare the weekly and monthly strategies. It's a great way to help users figure out which strategy fits their investment style and goals without getting bogged down in too much detail.

## 6. Testing and Evaluation

### 6.1 Testing

#### 6.1.1 Frontend

In web development, ensuring the functionality and user experience of the frontend is crucial. For this purpose, we utilise Cypress, a powerful tool for automating browser based tests. This allows us to simulate real user interactions directly within the browser, ensuring our application not only meets design specifications but also behaves as expected under various user scenarios.

A screenshot of a computer

Description automatically generated**Registration Form**

The registration form of Calzone Capital includes several user input validations critical for maintaining data integrity and compliance with business rules. Cypress automated the process of testing these validations to ensure they are correctly enforced. These tests cover checking mandatory field completion, validating the minimum password length, confirming age restrictions, and verifying the format of the phone number. Cypress scripts were crafted to sequentially fill out the form fields and verify each validation rule. This approach tests both the individual field validations and their dependencies, ensuring comprehensive coverage of the form's functionality. Successfully passing these tests confirms that the form is robust and behaves as expected, providing assurance of its readiness for real world use.

A screenshot of a computer

Description automatically generated**Login Form**

The login form of Calzone Capital plays a pivotal role in user authentication, demanding rigorous testing to ensure it functions correctly and securely. Cypress automated tests to validate the visibility of all required fields and the login button, ensuring users have all necessary components to log in. The tests also simulate the entry of user credentials, check for correct data capture, and verify the application's response to both correct and incorrect login attempts. This process helps confirm the robustness of the login form, its error handling capabilities, and the seamless navigation from the login to the registration page.

A screenshot of a computer

Description automatically generated**Navigation**

The navigation flow within Calzone Capital ensures users can smoothly move across different sections of the platform. Cypress tests verify that users can navigate from the landing page to both the registration and login pages, and from there to various key areas of the platform like the home, market, and profile pages. These tests also confirm the functionality of logging out, ensuring users are correctly redirected to the landing page, thus validating the integrity and security of user sessions and navigation paths within the application. This comprehensive testing ensures a user-friendly and secure navigational experience, enhancing user engagement and trust in the platform.

#### A screenshot of a black screen Description automatically generated6.1.2 Middleware (API Testing)

In the development of Calzone Capital, ensuring the reliability and responsiveness of the APIs is essential for seamless interaction between the frontend and the underlying data systems. For this purpose, Postman, a comprehensive tool for API testing, was utilised to simulate and verify the behaviour of various API endpoints under different conditions.

The API testing focused on validating the correctness of the API responses and their compliance with the expected HTTP status codes for a range of scenarios. For instance, successful requests for market news, stock searches, and investment strategies returned a status code of 200, indicating proper function. Conversely, error handling was tested by triggering requests that should fail, such as unauthorised access to resources, which correctly resulted in a 401 status code. Additionally, scenarios involving invalid parameters or unprocessable requests were examined, expecting specific error codes (422) that aid in debugging and improving API robustness.

This examination ensures that all components interact correctly, data integrity is maintained, and any potential issues are identified and resolved promptly, enhancing the overall reliability and user experience of the platform.

#### 6.1.3 Backend (Model Testing)

The model testing in the development of Calzone Capital is critical to ensure the accuracy and reliability of the financial predictions and strategies offered by the application. The backend, built with Flask and incorporating various machine learning and statistical analysis libraries, handles complex data processing and prediction tasks.

The machine learning model is rigorously tested by evaluating its performance on historical data. The training process involves using real stock market data, which is preprocessed using MinMaxScaler to normalise the input features for the neural network. The model's predictions are then compared against actual historical prices to compute error metrics such

1. **Mean Absolute Error (MAE)**: Measures the average error in predictions, providing a straightforward indication of prediction accuracy without direction (over or under prediction).
2. **Mean Squared Error (MSE)**: Represents the average of the squares of the errors. MSE emphasises larger errors more than smaller ones, making it sensitive to outliers in prediction.
3. **Root Mean Squared Error (RMSE)**: The square root of MSE, which scales the errors back to the original units of the data, making it interpretable and comparable to the target variable's scale.
4. **R2 Score**: Indicates how well the predictions approximate the actual data points. An R2 score of 1 means perfect prediction accuracy, while a score closer to 0 indicates less accurate predictions.

These metrics help quantify the model's performance, showing how closely the model's predictions match the actual historical data. They are crucial for identifying the strengths and weaknesses of the predictive models used in the platform.

   # Calculate evaluation metrics

   mae = mean\_absolute\_error(actuals, predictions\_adjusted)

   mse = mean\_squared\_error(actuals, predictions\_adjusted)

   rmse = sqrt(mse)

   r2 = r2\_score(actuals, predictions\_adjusted)

   # Prepare and return the evaluation data

   evaluation\_data = {

       "MAE": mae,

       "MSE": mse,

       "RMSE": rmse,

       "R2": r2,

   }

#### 6.1.4 Usability Testing

Usability testing is a crucial component of ensuring that Calzone Capital not only meets functional requirements but also provides a user-friendly experience. This testing involves real users interacting with the application to uncover flaws and gather actionable feedback.

The testing was conducted by inviting a diverse group of individuals who represent the target user base (novice investors). Participants were asked to complete a series of tasks typically performed by users of the platform, such as registering an account, navigating through different sections, using the predictive model to help and support decision making.

**Key Findings**

1. **Performance and Responsiveness**: Users appreciated the quick responses of the application during navigation but noted that the loading times for graphical data representations were longer than expected, suggesting the need for optimization.
2. **Accessibility Features**: Feedback indicated that the application could improve in terms of accessibility, especially for users with visual impairments. Suggestions included better contrast, larger fonts, and screen reader compatibility.

Based on the feedback, several enhancements are recommended:

* **Optimising Load Times**: Despite efforts to improve loading times, certain limitations due to hardware performance were identified. To mitigate user frustration associated with longer load times, a loading screen with engaging music was introduced. This not only enhances the user experience by making the wait more pleasant but also creates a perception of faster loading.
* **Enhancing Accessibility**: Implementing standard web accessibility guidelines to accommodate all users, including those with disabilities.

Usability testing has proven invaluable in understanding how real users interact with Calzone Capital. The insights gained have directly informed several iterations of the user interface design, ensuring that the application is not only powerful in its functionality but also delightful and straightforward to use.

### 6.2 Evaluation of Testing Strategies

The evaluation of testing strategies across different components of Calzone Capital demonstrates a robust approach to ensuring software quality and reliability. Here’s a brief assessment:

**Frontend Testing**

Frontend testing with Cypress has proven to be effective in validating both the functional aspects and user experience of the application interface. By automating interactions with the user interface, we have ensured that the application behaves as expected across various scenarios, significantly reducing the risk of UI regressions and enhancing user satisfaction.

**Middleware Testing**

API testing with Postman has enabled us to verify the application's backend responsiveness and the integrity of data exchanges between the frontend and the server. This testing ensures that the application can handle expected and unexpected user inputs reliably, manage errors, and maintain data integrity, thereby supporting seamless and secure operations.

**Backend Testing**

The backend model testing has been crucial in validating the predictive capabilities of the application. Testing using historical data and evaluation metrics like MAE, MSE, RMSE, and R2 Score, we have been able to measure the accuracy of financial predictions and refine the model accordingly. This rigorous validation ensures that the model remains reliable and provides valuable insights for users.

### 6.3 Challenges and Limitations

While the testing strategies have been largely effective, certain challenges and limitations were encountered:

* **Data Dependency**: The performance and reliability of predictions are highly dependent on the quality and quantity of historical data available. Any discrepancies in data can significantly affect the outcomes.
* **Testing Scope**: While automated tests cover a significant portion of scenarios, there might still be edge cases that are not accounted for due to the vast possibilities of user interactions and data inputs.
* **Integration Complexity:** Testing the integrations between different components of the system poses its own set of challenges. Each component may function as expected individually but could fail when integrated if the interfaces between them are not properly defined or implemented.
* **Real-Time Data Handling:** The application's ability to handle real-time data and provide up-to-date financial advice is crucial. However, testing this capability is challenging due to the variability of financial data. Ensuring that the system performs well under different data loads and updates is a key area that requires ongoing attention.
* **Environmental Variability:** The application must perform consistently across different operating systems, browsers, and devices. This variability can introduce inconsistencies in how application features are experienced by users. Cross-platform testing is essential but can be resource-intensive and difficult to automate fully.

### 6.4 Future Directions

To further enhance the reliability and robustness of Calzone Capital, future testing efforts could focus on:

* **Increased Test Coverage**: Expanding the test scenarios to cover more edge cases and rare data conditions.
* **Performance Testing**: Implementing performance tests to ensure that the application scales well with an increasing number of users and data volume.
* **Continuous Integration/Continuous Deployment (CI/CD)**: Integrating automated tests into a CI/CD pipeline to ensure ongoing quality assurance throughout the development lifecycle.

## 7. Critical Appraisal

### 7.1 Project Execution Analysis

In assessing the project's progression against the initial Gantt chart created in November 2023, notable deviations in task durations highlight the challenges of software development project management. For example, the development of the dashboard page, initially planned to be completed within a week, extended significantly beyond its estimated timeframe, concluding in March 2024. This extension illustrates the underestimation of the technical and functional complexities involved.

A screenshot of a project management

Description automatically generatedA screenshot of a project management

Description automatically generated

Actual

Planned

In reviewing the progression of the dashboard page development against the initial plans, it's evident that the timeline was significantly extended due to ongoing adjustments that were directly influenced by the backend development, particularly the predictive model. The dashboard's functionality and presentation were continuously adapted to handle the incoming data from the backend effectively. This iterative process was necessary to ensure that the dashboard could accurately display the predictive analytics in a user-friendly manner. These adjustments were not only technical but also design-oriented, aiming to enhance user engagement and comprehension. The extended development time, while impacting the project timeline, also allowed for a more refined and user centric final product, demonstrating the balance between adhering to schedules and accommodating necessary quality improvements.

A screenshot of a computer

Description automatically generated  
A screenshot of a project management

Description automatically generated

Actual

Planned

The second semester Gantt charts reveal significant strategic shifts and adjustments. Initially, the semester was ambitiously scheduled to include a wide range of complex tasks, notably in areas such as algorithm development for stock price prediction and portfolio optimization. However, the chart depicting the actual outcomes shows a notable simplification and reduction of tasks. This adjustment indicates a focused approach where certain features were deprioritized and merged into broader categories to enhance project manageability and ensure the integrity of the core functionalities.

Moreover, the actual outcomes chart demonstrates extended durations for several key tasks, particularly in frontend development. This extension was necessary to accommodate continuous refinements and testing, ensuring that the final product met user expectations and system performance standards.

### 7.2 Achievements and Shortcomings

**Achievements**

The project successfully integrated a robust suite of functionalities that significantly enhanced user engagement and provided tools for stock market analysis and decision making. These features, ranging from real-time data updates to predictive analytics, have collectively elevated the platform's utility and have been instrumental in delivering a sophisticated user experience.

**Shortcomings**

Despite the successes, several planned features were not fully implemented due to various constraints:

* **Buying and Selling Stocks**: Initially planned as a fun and interactive feature, the functionality to directly buy and sell stocks was deprioritised and ultimately not implemented. It was intended to enhance the user experience by simulating a real trading environment but stood lower on the priority list.
* **Portfolio Optimisation**: This feature would allow users to manage multiple portfolios and monitor their value changes, was recognised as highly complex and was the least priority. It remained unimplemented as it required significant time and resources that were not feasible.
* **Wallet Page and Settings Page**: The wallet page, intended for adding credits and the settings page were started but never fully completed. They were both omitted from the final implementation. The settings page, in particular, was deemed less critical as the project progressed, leading to its development being halted.

### 7.3 Future Directions

Reflecting on this project's execution, I recognize the importance of more effective project planning for future endeavours. A critical lesson learned is the need to prioritise features more strategically and realistically assess the time required for each task. Moving forward, adopting a more streamlined approach to prioritisation will be essential. This means rigorously evaluating the impact and necessity of each feature before allocating resources, ensuring that vital functionalities that offer the most value to the user are developed first.

Moreover, enhancing the estimation process for project timelines will be crucial. Misjudging the duration needed for complex features, like the dashboard development, led to extended timelines and resource reallocation. In future projects, I plan to allocate more time upfront to understand the intricacies of each feature, possibly through preliminary research or consulting with more experienced developers. This approach will help in setting more accurate timelines and avoiding the cascading delays experienced in this project.

By focusing on these areas, I aim to improve the management and execution of future software development projects, ensuring a more efficient process and higher quality outcomes.

### 7.4 Discussion of Social, Sustainability, Commercial, and Economic Context

#### Social Impact

Calzone Capital aims to make advanced stock market tools accessible to everyday investors, previously available only to professionals. This can enhance financial literacy and enable fairer market participation. However, there's a risk that users might depend too heavily on the platform's advice without fully understanding potential risks, which could lead to financial losses.

#### Sustainability Impact

The project focuses on efficient data processing to minimise resource usage, aligning with sustainable technology practices. As the platform grows, it will be important to manage its energy consumption and carbon footprint, possibly by optimising algorithms and considering greener hosting solutions.

#### Commercial and Economic Context

The functionalities of Calzone Capital, like predictive analytics, have clear commercial potential. They could be integrated into existing trading platforms or offered as a new service to small investors and firms. This could disrupt traditional market dynamics by levelling the playing field and might attract investment from those interested in fintech innovations. Additionally, it could generate revenue through subscriptions or premium features.

#### Academic and Educational Relevance

In educational settings, this project serves as a practical example of applying machine learning and data processing in finance. It can be used in finance, computer science, or data science courses to demonstrate real-world applications of these technologies. The project’s development process provides valuable insights for project management and software engineering courses, emphasising the practical challenges of technology development.

### 7.5 Personal Development Assessment

Throughout this project, I experienced significant growth in both my technical skills and understanding of advanced programming concepts. One of the major challenges was mastering Long Short-Term Memory (LSTM) models for predictive analytics. This involved not only learning about the theoretical underpinnings of LSTM networks but also applying them practically to forecast stock prices, which enhanced my machine learning expertise.

Additionally, the project required me to develop proficiency in NextJS and Tailwind CSS, areas in which I had limited prior experience. Through extensive learning and application, including tackling numerous tutorials and practical implementations, I significantly improved my web development skills. These experiences were pivotal in enhancing my abilities to integrate complex functionalities into user-friendly web interfaces. These technical skills are highly relevant to my future career aspirations in software development and data science. The proficiency gained in both frontend and backend development, coupled with a deep understanding of machine learning applications, positions me well for roles that demand technical competencies in these areas. This project has not only broadened my technical skill set but also increased my confidence in handling complex software projects, preparing me for future challenges in the tech industry.

## 8. Conclusion

Calzone Capital stands as a significant step forward in making the complexities of stock market investments accessible to novice investors. By integrating tools like real-time stock updates, market news, and predictive analytics, the platform simplifies the investment process. The successful implementation of these features demonstrates the project's ability to meet its primary goals, enhancing user confidence and decision making in the stock market.

Throughout the development process, valuable insights were gained in areas such as project management, scope definition, and user-centred design. The project underscored the importance of meticulous planning and iterative testing, especially when dealing with sophisticated technologies and user interfaces. Moving forward, the project could benefit from additional features and further refinement of existing tools to boost functionality and user engagement. Future work will also focus on ensuring the platform's adaptability to changing market conditions and technological advancements.

This dissertation not only summarises the achievements and challenges of the Calzone Capital project but also sets the stage for ongoing improvement and potential research in financial technology. It highlights the project's role in educating novice investors and its potential to influence future developments in the field.

## 

## References

[1] Langager, C. (2023). How to Invest in Stocks: A Beginner’s Guide. *Investopedia*. [Online], Available: <https://www.investopedia.com/articles/basics/06/invest1000.asp>.

[2] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation.

[3] Gers, F.A., Schmidhuber, J., & Cummins, F. (2000). Learning to Forget: Continual Prediction with LSTM.

[4] Piotroski, J.D. (2000). Value Investing: The Use of Historical Financial Statement Information to Separate Winners from Losers

[5] Bollinger, J. (2001). Bollinger on Bollinger Bands.

[6] Lee, C.M.C., & Ready, M.J. (1991). Inferring Trade Direction from Intraday Data.

[7] eToro. (2023). About Us. [Online]. Available: https://www.etoro.com/about/.

[8] Hilpisch, Y. (2014). Python for Finance. *O’Reilly*.

[9] Udacity. (2022). Machine Learning for Trading. *Udacity*. [Online], Available: <https://www.udacity.com/course/machine-learning-for-trading--ud501>.

[10] Academind by Maximilian Schwarzmüller. (2023). Full Modern React Tutorial (with Redux, NextJS & TypeScript). YouTube. [Online]. Available: <https://www.youtube.com/playlist?list=PLXmMXHVSvS-DvYrjHcZOg726m3G9on0aD>

[11] Brownlee, J. (2021). Deep Learning for Time Series Forecasting. Machine Learning Mastery.

[12] Tailwind Labs. (2023). Tailwind CSS: From Zero to Production. [Online]. Available: <https://tailwindcss.com/docs>

[13] Twelve Data. "API Documentation," Twelve Data, 2023. [Online]. Available: <https://twelvedata.com/docs#reference-data>.

[14] Finnhub. "API Documentation," Finnhub, 2023. [Online]. Available: <https://finnhub.io/docs/api>.