



MarketMateFX

Final Project Report

TU856
BSc in Computer Science

Yaroslav Hrabas

C20394791

Supervisor: Damian Bourke

School of Computer Science
Technological University, Dublin

12/04/2024

Abstract

Forex trading has long benefited from advanced data analysis, but this advantage has typically been reserved for traders at higher levels due to the complexity and cost of advanced trading tools. MarketMateFX aims to open access to sophisticated trading analytics by providing a comprehensive trading journal application accessible to traders at all levels. This project focuses on the development of an intuitive and scalable platform that integrates modern technologies such as Node.js, ReactJS, and Firebase to ensure real-time processing, secure data management, and a user-friendly experience.

The main goal of this project is to create an application that not only tracks and analyzes trading performance but also enhances traders' decision-making abilities through detailed insights and trend analysis. Throughout the development process, challenges related to scalability, security, and user engagement were addressed to ensure that MarketMateFX could provide reliable and actionable information to its users.

MarketMateFX is built as a web-based application that facilitates easy access and provides a range of analytical tools, from basic trade recording to complex performance metrics. The end result is a platform that leverages detailed data visualizations and user-centric design to empower Forex traders to optimize their strategies and improve their trading outcomes. The application is a significant step towards leveling the playing field in Forex trading by making high-end tools available to a broader audience.

Declaration

I hereby declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed:

Yaroslav Hrabas

Student Name

Date: 12/04/2024

Acknowledgements

A huge thank you to my project supervisor Damian Bourke for helping me throughout the year on this project especially when it came to times where I was not always available. He always came correct with great answers to my questions.

Thank you to all the testers who put up with me constantly sending them new components to test and who gave me great ideas to add to my website.

I would also like to thank all other people who contributed to the success of my website no matter what the capacity was.

Was an absolute pleasure to work on this.

Table of Contents

1. Introduction.....	7
1.1. Project Background.....	7
1.2. Project Description.....	7
1.3. Project Aims and Objectives.....	7
1.4. Project Scope.....	7
1.5. Thesis Roadmap.....	7
2. Literature Review.....	8
2.1. Introduction.....	8
2.2. Alternative Existing Solutions.....	8
2.3. Technologies Researched.....	8
2.4. Other Relevant Research.....	8
2.5. Conclusions.....	8
3. Experiment / Software Design.....	9
3.1 Introduction.....	9
3.2. Software Methodology.....	9
3.3. Requirements Gathering.....	9
3.4. Overview of System.....	9
3.5. Wireframes.....	9
3.6. Conclusions.....	9
4. Experiment / Software Development.....	10
4.1. Introduction.....	10
4.2. Software Development.....	10
4.3. Problems that arose during Development	10
4.4. Conclusions.....	10
5. Testing and Evaluation.....	11
5.1. Introduction.....	11
5.2. System Testing.....	11
5.3. System Evaluation.....	11
5.4. Conclusions.....	11
6. Conclusions and Future Work.....	12
6.1. Conclusions.....	12
Bibliography.....	13

Table of Figures

Figure.1 - Sample Forex Candle Chart
Figure.2 - Tradervue Dashboard
Figure.3 - TraderSync Dashboard
Figure.4 - Edgewonk Dashboard
Figure.5 - Example of MIT license
Figure.6 - Scrum Process
Figure.7 - Waterfall Method
Figure.8 - ERD
Figure.9 - Case Diagram
Figure.10 - Class Diagram
Figure.11 - System Architecture
Figure.12 - Login Wireframe
Figure.13 - Dashboard Wireframe

Figure.14 - Code Snippet of trade form

Figure.15 - Add Trade Form

Figure.16 - Risk Function

Figure.17 - Risk Component

Figure.18 - Win/Loss Code

Figure.19 - Win/Loss Component

Figure.20 - Authentication Code

Figure.21 - MarketMateFX Dashboard

Figure.24 - MarketMateFX Login

Figure.25 - Test Code

Figure.26 - List of User Profiles

1.1. Project Background

The foreign exchange market (Forex) is the largest and most liquid financial market in the world, with a daily trading volume exceeding \$6 trillion as of 2019 (1). This vast market offers immense opportunities for traders, but also presents unique challenges, particularly in terms of decision-making and strategy optimization. While professional traders often have access to sophisticated tools and data analysis, amateur and semi-professional traders typically do not have the same resources at their disposal, potentially putting them at a significant disadvantage.

MarketMateFX is a solution to bridge this gap. Developed as a comprehensive trading journal application, MarketMateFX is designed to enhance the trading experience at all levels by providing advanced analytics, real-time data processing, and user-friendly features. This project harnesses the power of modern technologies including Node.js for backend operations, ReactJS for a dynamic

frontend, and Firebase for seamless data management and authentication. The integration of these technologies ensures that MarketMateFX is not only robust and scalable but also accessible anywhere by anyone interested in Forex trading.

This report outlines the development process of MarketMateFX, highlighting the challenges encountered and the solutions implemented to address them. It details the project's scope, design considerations, technological architecture, and the benefits it offers to users. The ultimate end goal of MarketMateFX is to make these trading tools accessible to a wider audience and therefore leveling the playing field in the Forex trading community.

1.2. Project Description

MarketMateFX is a visionary project dedicated to developing an advanced and user-centric trading journal application exclusively designed for Forex (FX) traders. The primary aim of this application is to empower traders of all experience levels with a comprehensive toolset for effectively tracking, analysing, and enhancing their trading performance. At the heart of MarketMateFX is a plan to simplify the very intricate process of managing trade data and related information while ensuring a secure and intuitive trading environment. The key project objectives encompass the creation of an intuitive and seamless, friendly user interface, robust data capture and storage mechanisms, in-depth performance analytics, risk management tools, a dynamic notification system, insightful generated reports, data import and export capabilities, robust security measures for data protection, cross-platform accessibility, and extensive user support resources.

The anticipated benefits of MarketMateFX can scale infinitely. This trading journal promises to be a game-changer in the world of Forex by offering traders valuable insights into their trading history, facilitating data-driven decision-making, and promoting efficiency in organising different aspects of trading. The user-friendly interface will elevate the overall trading experience, simplifying data recording and analysis. Incorporating different risk management features and strategies, MarketMateFX will empower and ensure traders maintain discipline and minimise potential losses in their trades. In conclusion, MarketMateFX aspires to redefine FX Trading by becoming a tool that every trader will be using to ensure their trading success.



Figure.1 - Sample Forex Candle Chart

1.3. Project Aims and Objectives

Overall Aim

The overall aim of MarketMateFX is to revolutionize the Forex trading experience by providing a comprehensive, user-friendly, and analytical trading journal tool that empowers traders of all experience levels to optimize their strategies, improve decision-making, and enhance overall trading performance.

Milestones

- Development Completion: Finalize the core development of MarketMateFX, ensuring all planned features are fully functional and user-tested.
- User Acquisition: Successfully onboard the first 10 active users, indicating initial market penetration and user interest.
- Feedback Integration: Implement significant improvements based on initial user feedback, enhancing usability and functionality.

Objectives

The objectives of MarketMateFX can be detailed as follows, each contributing towards reaching the milestones and the overall aim:

- Develop an Intuitive User Interface: Create an easy-to-navigate, visually appealing interface that simplifies trade logging, analysis, and review for traders.
- Implement Robust Data Analytics: Offer comprehensive analytics tools that allow users to dissect their trading performance across various metrics, identifying strengths and areas for improvement.

- **Incorporate Advanced Risk Management Tools:** Provide users with tools to calculate and manage risk effectively, including stop-loss calculators, risk-reward ratio analyses, and personalized risk profiles.
- **Ensure Cross-Platform Accessibility:** Develop the application for accessibility on multiple platforms (web, mobile iOS, and Android), ensuring seamless synchronization and usability across devices.
- **Maintain Highest Standards of Data Security:** Implement top-tier encryption and security protocols to protect user data, ensuring trust and compliance with global data protection regulations.
- **Regularly Update and Innovate the Feature Set:** Commit to continuous improvement and the introduction of new features based on technological advancements, market trends, and user feedback.

1.4. Project Scope

Project Scope for MarketMateFX

In Scope:

- **Trading Journal Functionality:** Core development of a trading journal application designed for Forex traders to log, track, and analyze their trades.
- **User Interface (UI) and User Experience (UX):** Development of an intuitive and user-friendly interface that caters to traders of all experience levels.
- **Performance Analytics:** In-depth analytics tools that provide insights into trading performance, including profit and loss analysis, win/loss ratios, and trade patterns.
- **Security:** Implementation of robust security measures to protect user data, including encryption and compliance with data protection laws.
- **Cross-Platform Support:** Development for multiple platforms, ensuring compatibility with web, iOS, and Android devices.

Out of Scope:

- **Brokerage Services:** MarketMateFX will not function as a brokerage platform, nor will it facilitate direct trading or financial transactions.

- Financial Advice: The application will not provide personalized financial advice, recommendations for trades, or investment strategies.
- Market Data: Live market data feeds, financial news, or in-depth market analysis will not be primary features of MarketMateFX. While the tool may allow for basic data import/export, it will not serve as a comprehensive source for real-time financial information.
- Automated Trading: MarketMateFX will not support automated trading systems or the execution of trades based on predefined criteria.
- Portfolio Management: Comprehensive portfolio management features, such as asset allocation or diversification strategies, will not be included. The focus will remain on trade journaling and analysis.

This scope shows and ensures that MarketMateFX remains focused on its goal to enhance the Forex trading experience through effective trade journaling and analysis, while setting clear boundaries on what users can expect from this application.

1.5. Thesis Roadmap

The structure of this thesis report follows the progress that was made during the development of this project. The report starts with the introduction including the background of the project and what are the expected objectives of this project.

The literature Review/Research section aims to give an insight into forex trading and how technologies can be combined together to bring a website to life. This section presents the advantages and disadvantages of the possible technologies.

Experiment/Software Design showcases the several ways of requirements gathering that was conducted prior to the development of MarketMateFX.

The Software Development section is used to showcase the various parts of the website as well as the completed website UI and any problems that were encountered.

The testing and evaluation section contains details on the testing that was conducted and evaluated in order to have the website working at top performance.

2. Literature Review

2.1. Introduction

This report presents an analysis of MarketMateFX, an advanced trading journal designed for Forex (FX) traders. The objective of this report is to provide an overview of the platform's key features, functionality, and its potential impact on the world of FX trading.

The foreign exchange market, commonly known as Forex or FX trading, is a dynamic and global arena where currencies are bought and sold (1). Unlike traditional financial markets, Forex trading operates without a central marketplace, and transactions are executed electronically over the counter (OTC) through a network of interconnected computers worldwide. This unique feature allows the Forex market to function 24 hours a day, five and a half days a week, spanning major financial centres across different time zones.

The Forex market is often characterised by terms like FX, foreign exchange market, or currency market, all of which are used interchangeably (2).

Forex trading has evolved over time, transitioning from being dominated by institutional firms and large banks to becoming more inclusive, with traders and investors of various sizes participating in the market. Unlike traditional stock exchanges or commodities markets, there are no physical trading venues for Forex. Instead, it operates as a decentralised network of connected trading terminals and computer networks. Market participants encompass institutions, investment banks, commercial banks, and retail investors from around the world. It is a decentralised global market meaning that it is not controlled by any one central authority (3). This allows anybody to jump into FX trading at will.

Forex trading occurs primarily through spot, forwards, and futures markets, with the spot market being the largest with an average daily turnover of \$5 trillion (4). The spot market is where currencies are exchanged based on their prevailing trading prices, determined by factors like interest rates, economic performance, geopolitical sentiment, and price speculation. A spot deal involves a bilateral transaction in which one party delivers a specified currency amount to the counterparty and receives another currency at the agreed-upon exchange rate. These transactions typically take two days to settle.

The forwards and futures markets, on the other hand, are favoured by entities like companies and financial firms seeking to hedge their foreign exchange risks. Forward contracts involve private agreements between two parties to buy a currency at a future date at a predetermined price (5). Futures contracts, traded on public exchanges, are standardised agreements for currency delivery at a future date and a predetermined price.

Forex trading offers unique opportunities for profit and risk management. Traders can capitalise on differences in interest rates between two currencies or profit from fluctuations in exchange rates. Strategies like carry trades, which involve taking advantage of interest rate differentials between currencies, have been employed by traders for profit.

The Forex market is utilised by companies conducting business across international borders to mitigate currency risk. These companies can lock in exchange rates through forward or swap markets, ensuring predictability in their transactions.

In this world of Forex trading, MarketMateFX plans to emerge as an advanced trading journal designed to simplify trading processes, empower traders with data-driven insights, and enhance the overall trading experience. It provides traders with a user-friendly interface, tools for trade documentation, performance analytics, and risk management features

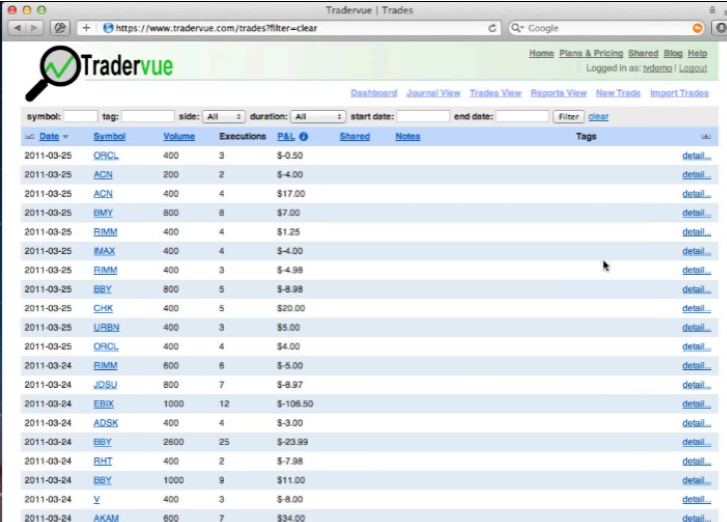
MarketMateFX is in a position to address traders' needs for effective record-keeping and performance assessment in the dynamic Forex market. This report will dive into MarketMateFX's features, functionality, and its potential impact on Forex trading. In today's world of FX, it is especially important to document all trades in a journal for maximum efficiency and educational purposes.

2.2. Alternative Existing Solutions

There are many Trading Journals due to the importance of documentation and analysis of trading in the trading world but these journals look after all kinds of trading which can be overwhelming sometimes. Discussed down below, are the advantages and the disadvantages of each application and how they provide their own benefits when it comes to the documentation of trades. Reviewing these similar applications will give a better understanding of what needs to be implemented and how for the successful creation of this Forex Trading Journal.

2.2.1. TraderVue:

Created in early 2011 by Greg Reinacker (6), TraderVue is a web-based application designed for traders of any tool or commodity. Upon its original launch, it actually did focus on mainly just Forex trading but then TraderVue was bought out by a company called SureSwift Capital. Since this company changed, TraderVue was then continuously developed to be a journaling platform for all kinds of trading from stocks to cryptocurrencies, etc. Now it serves as a fully comprehensive trading journal and analytical tool that helps traders track and analyse all their trading activities and learning from them



Date	Symbol	Volume	Executions	P&L	Shared	Notes	Tags	Actions
2011-03-25	ORCL	400	3	\$-0.50				detail...
2011-03-25	ACN	200	2	\$-4.00				detail...
2011-03-25	ACN	400	4	\$17.00				detail...
2011-03-25	BMV	800	8	\$7.00				detail...
2011-03-25	PLMM	400	4	\$1.25				detail...
2011-03-25	IMAX	400	4	\$-4.00				detail...
2011-03-25	PLMM	400	3	\$-4.98				detail...
2011-03-25	BBY	800	5	\$-8.98				detail...
2011-03-25	CHK	400	5	\$20.00				detail...
2011-03-25	URBN	400	3	\$5.00				detail...
2011-03-25	ORCL	400	4	\$4.00				detail...
2011-03-24	PLMM	600	6	\$-5.00				detail...
2011-03-24	JDSU	800	7	\$-8.97				detail...
2011-03-24	EBIX	1000	12	\$-106.50				detail...
2011-03-24	ADSK	400	4	\$-3.00				detail...
2011-03-24	BBY	2500	25	\$-23.99				detail...
2011-03-24	PLMT	400	2	\$-7.98				detail...
2011-03-24	BBY	1000	9	\$11.00				detail...
2011-03-24	V	400	3	\$-8.00				detail...
2011-03-24	AKAM	600	7	\$34.00				detail...

Figure.2 - Tradersvue Dashboard

Advantages of TraderVue:

- TraderVue gives traders a massive range of different ways to detail and note their trades. Users can record information about their trades, including entry and exit points, trade

duration, currency pairs traded, and trading strategies used. This record-keeping is vital for traders looking to improve their trading skills by analysing past performances.

- The application offers a range of performance analytics tools, allowing traders to assess their trading strategies and profits. It provides visual representations of trading data, helping traders identify trends, strengths, and weaknesses in their approach.
- TraderVue includes features to help traders manage risk effectively. Users can set risk tolerance levels and receive notifications if they exceed predefined risk parameters. Particularly useful for minimising potential losses.
- TraderVue allows traders to connect with other members of the trading community. Users can share their trading results, discuss strategies, and learn from each other's experiences. This social aspect can be beneficial for traders seeking advice and mentorship.
- As a web-based platform, TraderVue is accessible from various devices with an internet connection. Traders can log in and access their trading data from anywhere, making it convenient for those who are constantly on the move.

Disadvantages of TraderVue:

- Due to the amount of tools and features, the application has a very big learning curve for new users. Navigating the platform and fully understanding all its capabilities can take some time and effort.
- TraderVue's extensive features and analytics may be overwhelming for novice traders. The application's complexity might deter those who are just starting their trading journey.
- While there may be a free version of TraderVue available, some of its advanced features may require a paid subscription.

2.2.2. TraderSync:

Made in 2015 by a team overlooked by David Olivares (7), TraderSync is another web-based application designed for traders, with a focus on performance analysis and journaling. Upon initial launch, a lot of the features were based on investments which can just be considered very long term trades. Since its launch in 2015, TraderSync has developed to be a widely comprehensive journaling tool with many different features to accommodate all kinds of asset trading.

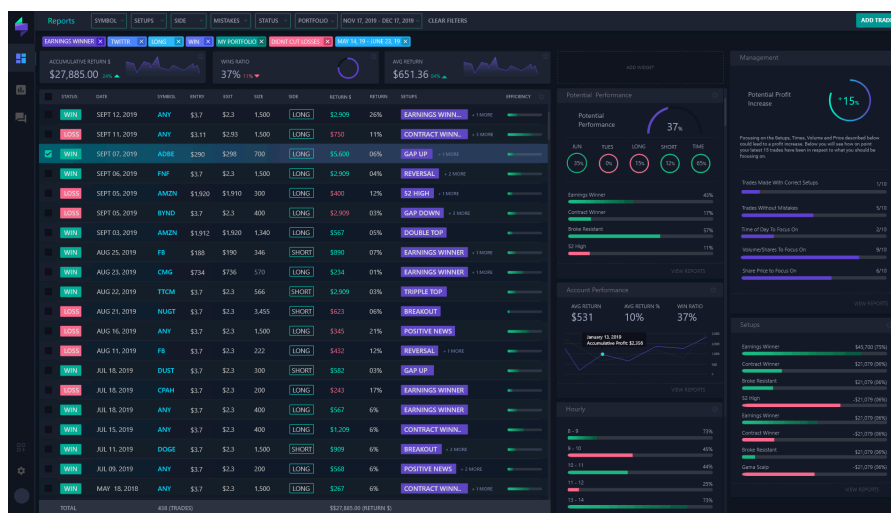


Figure.3 - TraderSync Dashboard

Advantages of TraderSync:

- TraderSync offers a wide range of performance analytics tools to help traders evaluate their trading strategies and results. Users can gain insights into their trading performance through various charts, statistics, and reports.
- Similar to TraderVue, TraderSync emphasises risk management. Users can set risk parameters, including maximum drawdown and risk per trade. The platform provides alerts and notifications to help traders stay within their predefined risk limits.
- This application boasts an intuitive and user-friendly interface. Traders can easily input their trade data, and the platform automatically calculates various performance metrics. The clear and organised layout makes it accessible for traders of all experience levels.
- TraderSync accommodates a broader range of asset classes, including stocks, options, cryptocurrencies, etc. This versatility makes it suitable for traders with diverse trading and investment portfolios.

Disadvantages of TraderSync:

- TraderSync offers both free and paid subscription plans. Some of the more advanced features and tools may require a premium subscription, which could be costly for some traders.
- Although TraderSync's interface is user-friendly, traders new to the platform may still require some time to fully grasp its features and functionalities as these features need to be applied independently to a different range of assets.
- This application is very limited in its automation. TraderSync focuses on manual trade entry and tracking, which may require more effort from traders who prefer automated data import from brokers or trading applications.

- Some users have reported mixed experiences with TraderSync's customer support. While the platform offers resources like video tutorials and FAQs, users may encounter challenges if they require direct assistance.

2.2.3. Edgewonk:

Edgewonk is a fairly advanced trading journal and performance analysis software designed for traders across various financial markets. Due to its very heavy functionality and also automation, it is actually only a paid software and does not offer any free version (8). Since its release, edgewonk actually has gone through many iterations and is now considered Edgewonk 3.0 offering more automation features and functionality with each version.

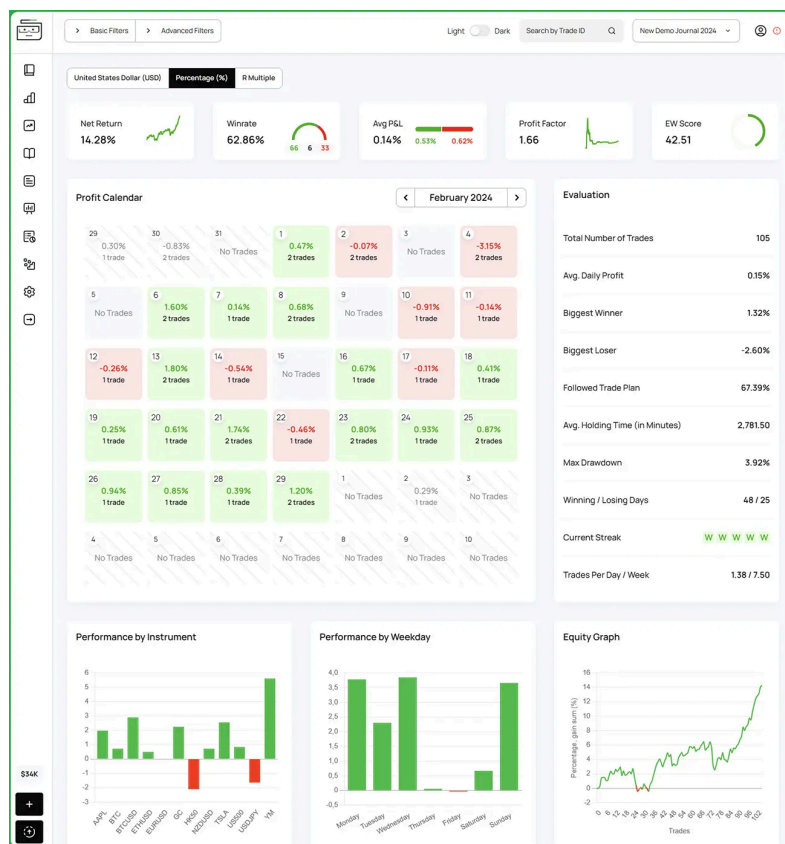


Figure.4 - Edgewonk Dashboard

Advantages of Edgewonk:

- Edgewonk is a very big comprehensive journal that allows traders to record and analyse their trades in detail. Users can input trade data, including entry and exit points, trade duration, position size, and trade management strategies.
- The platform provides extensive performance analysis tools, including various charts, statistics, and reports. Traders can gain insights into their trading habits, strengths, and weaknesses, helping them make data-driven improvements.

- Edgewonk goes beyond traditional trading journals by incorporating psychological analysis. It helps traders assess their emotions and discipline during trading, aiming to improve mental and emotional aspects of trading.
- Edgewonk allows users to customise their trading journal to suit their specific needs. Traders can create custom tags, categories, and statistics to track the aspects of trading that matter most to them.

Disadvantages of Edgewonk:

- Edgewonk is a paid software, and the cost may be a consideration for traders, especially those who are just starting and have limited trading capital.
- While Edgewonk is user-friendly, traders new to the platform may need some time to fully understand and utilise all its features effectively, especially if they want to use automation features.
- Edgewonk is suitable for traders involved in various asset classes, including stocks, Forex, and futures. However, traders exclusively involved in specific markets may find platforms tailored to those markets more suitable.
- Unlike other platforms, Edgewonk does not emphasise community or social interaction. It primarily focuses on individual trader analysis and improvement.

2.3. Technologies Researched

Several different technologies can be used for a project such as MarketMateFX. These technologies range from what can be used in relation to Front-end Frameworks, Back-end Frameworks, code languages and databases.

Each has their own uses and benefits so a careful selection process does need to take place to ensure the technologies used can cater to a forex trading journaling application.

2.3.1. Databases:

2.3.1a. Firebase Database:

Firebase (Realtime) Database is a cloud-hosted NoSQL database (9) provided by Google as part of the Firebase suite of application development tools. It is designed specifically for building real-time web and mobile applications, offering synchronisation and data storage in a JSON format.

Advantages of Firebase Realtime Database:

- Real-Time Data Synchronisation - Firebase Database excels in real-time data synchronisation. Any changes made to the database are immediately pushed to all connected clients, ensuring that users receive up-to-the-second updates without the need for manual refreshes. This feature is ideal for applications requiring live updates.
- Real-Time Querying - Firebase offers powerful real-time querying capabilities, allowing developers to filter and retrieve data in real-time based on specific criteria. This enables dynamic and responsive applications that adapt to user actions and preferences.

Disadvantages of Firebase Realtime Database:

- Structured Data Limitations - Firebase Database uses a JSON structure for data storage, which can be limiting for complex data models. Nested data structures can become challenging to query and manage as the application scales, potentially leading to inefficient database structures.
- Limited Query Capabilities - While Firebase supports real-time querying, its query capabilities are more limited compared to traditional SQL databases. Complex queries often require additional client-side processing, which can impact performance and development complexity.

2.3.1b. MongoDB:

MongoDB is a widely-used NoSQL database management document based (10) system known for its flexibility and scalability. It is a large document-oriented database and is designed to store and manage data in JSON-like BSON (Binary JSON) format.

Advantages of MongoDB:

- Flexible Schema - MongoDB offers a flexible schema design, allowing developers to store data without a fixed structure. This flexibility is particularly beneficial when dealing with evolving data requirements, as changes to the data model can be made without requiring a predefined schema.
- Scalability - MongoDB is built for horizontal scalability, making it suitable for applications with high data volumes and traffic. It can distribute data across multiple servers or clusters, allowing for seamless scaling as the application grows. Features like sharding and replica sets enhance its scalability.

Disadvantages of MongoDB:

- Complex Queries - While MongoDB's query language is powerful, complex queries can be challenging to write and optimize. Queries that involve aggregations, joins, or complex filtering may require additional effort compared to traditional relational databases.

- Data Consistency - MongoDB's default configuration prioritises performance over immediate data consistency. In certain scenarios, such as network partitioning or server failures, MongoDB may not provide strict data consistency by default. Developers must carefully design their data models and configure write concerns to ensure data consistency when needed.

2.3.1c. MariaDB:

MariaDB is a widely-used open-source relational database management system that is known for being a fork of MySQL. It was developed as an alternative to MySQL and is designed to be compatible with MySQL's syntax and database structure.

Advantages of MariaDB:

- Open-Source - MariaDB is an open-source relational database management system, which means it is freely available to use, modify, and distribute. This open-source nature makes it a cost-effective choice for organisations or individual developers.
- Compatibility with MySQL - MariaDB is designed to be highly compatible with MySQL, both in terms of SQL syntax and data structures. This makes it relatively easy for users who are familiar with MySQL to transition to MariaDB without major code or database changes. Additionally, many applications and libraries that work with MySQL also work seamlessly with MariaDB.

Disadvantages of MariaDB:

- Smaller Ecosystem - MariaDB has a smaller user and developer community compared to MySQL. This can result in fewer resources, plugins, and third-party tools available for MariaDB. While it is actively maintained, users may not find as extensive a support ecosystem as they would with some other database systems.

2.3.1d. PostgreSQL:

PostgreSQL, often referred to as "Postgres," is a powerful and open-source (11) relational database management system known for its robust features and extensibility.

Advantages of PostgreSQL:

- Advanced Features - PostgreSQL is known for its comprehensive set of advanced features, including support for complex data types, indexing techniques, and custom functions. It provides features such as full-text search, geospatial data support, and JSON support, making it suitable for a wide range of applications.

- **Extensibility** - PostgreSQL is highly extensible, allowing users to create custom functions, operators, and data types. This extensibility, combined with a vibrant and active open-source community, enables developers to enhance and tailor PostgreSQL to meet specific project requirements.

Disadvantages of PostgreSQL:

- **Learning Curve** - PostgreSQL, while powerful, can have a steeper learning curve for developers who are new to relational databases or who have experience with simpler systems. Its extensive feature set and configuration options may require time and effort to master fully.
- **Resource Intensive** - PostgreSQL can be more resource-intensive compared to some other database systems, especially in terms of memory and CPU usage. This can be a consideration for organisations or individual developers with limited hardware resources or those running on cloud platforms with resource constraints.

2.3.1e. Selection: Firebase Database

Firebase has been chosen as the database solution for this MarketMateFX project based on a thorough assessment of its features and how they align with the project's objectives and requirements. Firebase brings several key advantages to the table that make it an excellent fit for a trading journal application such as MarketMateFX.

One of Firebase's standout features is its real-time data synchronisation capability. In the context of MarketMateFX, this feature is invaluable. It ensures that any data updates or changes made by one user are instantly reflected across all clients. This real-time synchronisation is crucial for a trading application where timely access to accurate trading data is of utmost importance. It enables traders to always have access to the latest information, facilitating informed decision-making and trade analysis.

Additionally, Firebase provides robust data persistence, a critical aspect when deploying an application on the cloud. MarketMateFX's primary focus is on data security and reliability. Firebase ensures that users' trading records are securely stored and maintained across all clients, even in the case of network disruptions or client-side issues. This persistence guarantees that users' data remains intact and accessible, contributing significantly to the application's reliability.

MarketMateFX will implement various authentication mechanisms to ensure secure access control. Firebase streamlines the integration of these authentication mechanisms by offering a range of libraries and software development kits (SDKs) designed for client user authentication. This feature simplifies the implementation of secure user access, enhancing the overall security posture of the application.

Another critical requirement for MarketMateFX is the ability to support multiple file types for demonstrating access control effectively. Firebase meets this requirement by allowing the storage of various file types, including images and videos. This capability empowers MarketMateFX to showcase its access control mechanisms comprehensively, reinforcing user confidence in the application's security features.

These features that Firebase has to offer, aligns perfectly with the project's goal of creating a secure and user-friendly trading journal application.

2.3.2. Front-End Frameworks:

2.3.2a. ReactJS:

React.js is a popular and widely-used open-source JavaScript library for building user interfaces. Developed and maintained by Facebook, React has gained widespread adoption in web development due to its component-based architecture and efficient rendering system.

Advantages of React.js:

- **Component-Based Architecture** - React follows a component-based architecture, which allows developers to create reusable UI components. Each component can encapsulate its logic, state, and user interface elements. This modularity simplifies development, enhances code maintainability, and promotes a consistent and organised code structure.

Disadvantages of React.js:

- **Learning Curve** - React introduces several new concepts, such as JSX (JavaScript XML) for defining UI components and the unidirectional data flow. For developers who are new to these concepts, there can be a learning curve associated with mastering React.

2.3.2b. AngularJS:

Angular is a powerful front-end framework that offers a comprehensive set of tools and features for building web applications. It integrates seamlessly with TypeScript, providing static typing and improved code maintainability. However, its learning curve can be steep, and the framework may require developers to write significant amounts of boilerplate code, potentially impacting development efficiency.

Advantages of Angular:

- **Powerful Framework** - Angular is a comprehensive and feature-rich framework for building web applications. It provides a structured and opinionated approach to development, which can be advantageous for large-scale projects. Angular offers tools and libraries for tasks like routing, forms, and state management, reducing the need for external dependencies.
- **TypeScript Integration** - Angular is built with TypeScript, a statically-typed superset of JavaScript. TypeScript brings benefits like improved code maintainability, enhanced developer tooling, and early error detection. This strong typing helps catch errors during development and makes code easier to understand and refactor.

Disadvantages of Angular:

- **Learning Curve** - Angular has a steeper learning curve compared to some other front-end frameworks due to its complex architecture and the need to learn TypeScript. Developers may require more time and effort to become proficient in Angular.
- **Boilerplate Code** - Angular applications can involve writing a significant amount of boilerplate code, which may make development more verbose and time-consuming. While this boilerplate code can provide structure, it may also lead to increased development time and effort.

2.3.2c. Selection: ReactJS

React.js is specifically designed for building user interfaces in web applications.

In React, UIs are divided into modular and reusable components. These components encapsulate their own logic, state, and rendering. This approach offers several advantages for MarketMateFX:

React's component reusability is a significant benefit. Traders often need to interact with similar UI elements across various sections of the application, React allows for the creation of reusable components. This not only streamlines development but also ensures a consistent and user-friendly experience.

React's efficient rendering capabilities are crucial for real-time data updates which is a key requirement in a trading journal. React employs a Virtual DOM and an efficient diffing algorithm, minimising unnecessary DOM manipulations. This results in faster rendering and a highly responsive user interface, providing traders with up-to-the-minute information without delays.

React can be seamlessly integrated with TypeScript if desired. This integration offers the benefits of strong typing and enhanced development tooling, which can help prevent common programming errors and maintain code quality.

It is the chosen Front-end Framework for this project as it provides the necessary tools and features to create a responsive, user-friendly, and real-time trading journal application.

2.3.3. Back-End Frameworks:

2.3.3a. NodeJS

Node.js is an open-source, server-side JavaScript runtime environment that allows developers to run JavaScript code on the server. It is built on Chrome's V8 JavaScript engine and has gained widespread popularity for its versatility and efficiency in building scalable and high-performance web applications. Node.js is event-driven and non-blocking, making it particularly suitable for real-time applications and handling a large number of concurrent connections.

Advantages of Node.js:

- **Non-Blocking I/O** - One of Node.js's primary strengths is its non-blocking, event-driven architecture. This means that it can handle multiple I/O operations concurrently without waiting for one to complete before starting another. As a result, Node.js applications can efficiently manage real-time tasks, such as handling numerous incoming requests.
- **Large Ecosystem** - Node.js benefits from a vast and active ecosystem of open-source libraries and packages available through the Node Package Manager. This extensive collection of modules simplifies development by providing pre-built solutions for various functionalities.

Disadvantages of Node.js:

- **Callback Pyramid** - In asynchronous programming with Node.js, developers often encounter callback "hell", also known as the callback pyramid. This can make the code less readable and harder to maintain, as multiple nested callbacks can become complex. While various approaches and libraries exist to address this issue (such as Promises and `async/await`), managing asynchronous code effectively still requires careful planning and consideration. (*GeeksForGeeks, 2023*).

2.3.3b. Laravel (PHP)

Laravel is a popular open-source PHP web application framework. It is designed to simplify and accelerate the development of web applications, making it an attractive choice for a wide range of projects. It provides developers with a clean and enjoyable coding experience. It follows the Model-View-Controller (MVC) architectural pattern, helping developers structure their applications in a way that separates concerns and promotes code organisation.

Advantages of Laravel:

- Rich Ecosystem - Laravel has a robust and active ecosystem with a wealth of packages and extensions available through Composer, a PHP dependency manager. These packages cover a wide range of functionalities, from authentication and authorization to many others.
- Laravel features Eloquent, its own Object-Relational Mapping system (12), which simplifies database interactions. Eloquent allows developers to work with databases using expressive and intuitive syntax, making database operations more efficient and developer-friendly. It supports relationships, migrations, and query builder capabilities.

Disadvantages of Laravel:

- Learning Curve - While Laravel aims to simplify web development, it may still have a learning curve for developers who are new to the framework. Beginners may need time to grasp Laravel's conventions and concepts, especially if they are not familiar with PHP or MVC architecture.

2.3.3c. Selection: NodeJS

Node.js is an excellent choice for powering the backend of MarketMateFX, offering several advantages that align with the project's goals.

MarketMateFX may require real-time capabilities, such as live trading data updates and instant notifications. Node.js excels in this regard, thanks to its non-blocking, event-driven architecture. It efficiently handles real-time communication, making it ideal for implementing features that demand instantaneous updates, a crucial aspect of a trading application.

Scalability is another essential factor for MarketMateFX, as it should accommodate varying levels of user activity. Node.js is renowned for its scalability, capable of efficiently managing a broad range of concurrent users. Whether you have a few or thousands of users accessing the application simultaneously, Node.js can meet the demands of MarketMateFX effectively.

Node.js's cross-platform compatibility simplifies deployment, ensuring that MarketMateFX functions seamlessly across various hosting environments. Furthermore, its lightweight and efficient runtime contributes to excellent performance, a critical aspect for a trading application. Node.js can handle high concurrency and execute tasks swiftly, delivering the performance needed for MarketMateFX.

Additionally, Node.js integrates seamlessly with Firebase's JavaScript SDK, facilitating interactions with Firebase services such as the Realtime Database and Firestore. This integration simplifies data management and authentication, streamlining the backend development process for MarketMateFX.

2.3.4. Web Applications:

2.3.4a. PHP

PHP is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. (<https://www.php.net/manual/en/intro-what-is.php>) As PHP is heavily focused on server side scripting, this means it executes its HTML on the server first and then sends it to the client.

Advantages of PHP:

- **Ease of Learning** - PHP is relatively easy to learn, especially for developers with prior experience in programming languages like C, C++, or Java. Its syntax is straightforward and similar to other C-style languages.
- **Rapid Development** - PHP's simplicity and extensive libraries enable developers to build web applications quickly. This is particularly valuable for prototyping and launching projects quickly

Disadvantages of PHP:

- **Inconsistent Standard Library** - PHP's standard library can be inconsistent, with functions and features that vary in naming conventions and behaviour. This can lead to unexpected issues when working with different parts of the language.
- **Scalability Challenges** - While PHP is suitable for small to medium-sized web applications, it may face scalability challenges for large-scale projects.

2.3.4b. JavaScript

Java is a scripting or programming language that allows for the development and implementation of complex features on web pages (13).

It is very dynamic and versatile, therefore it plays a central role in web development. Unlike many other languages, JavaScript executes directly in a user's web browser, allowing it to enhance web pages with interactivity and dynamic behaviour. It is characterised by its high-level nature, making it relatively easy to read and write. JavaScript follows the ECMAScript standard, ensuring consistency and compatibility across different implementations.

Advantages of JavaScript:

- **Interactivity** - JavaScript enables the creation of dynamic and interactive user interfaces. It can respond to user actions in real-time, enhancing the user experience.

Disadvantages of JavaScript:

- **Browser Compatibility** - While modern JavaScript is well-supported in browsers, cross-browser compatibility can still be challenging.
- **Security Risks** - JavaScript can introduce security vulnerabilities, such as cross-site scripting attacks, if not handled correctly.

2.3.4c. Selection: JavaScript

JavaScript is a well-suited choice for developing MarketMateFX due to its versatile and interactive nature. This project requires a user-friendly interface that allows traders to input, track, and analyse their trades efficiently and JavaScript provides the means to develop that. JavaScript's dynamic capabilities enable real-time updates, responsive design, and seamless interactions, contributing significantly to an enhanced user experience.

MarketMateFX is a web-based application, and JavaScript is the primary language for client-side scripting in web development. It enjoys widespread support across modern web browsers, ensuring cross-browser compatibility. This means traders can access the application from various devices and browsers without any issues, expanding its reach to a much broader audience.

In terms of scalability and maintenance, JavaScript's flexibility allows structure of the codebase in a scalable and maintainable manner. This is vital for MarketMateFX, as the project may evolve over time, requiring updates and additional features. JavaScript's modularity and adherence to best practices make it easier to manage and extend the application should it be needed.

2.4. Other Relevant Research

Website Hosting

Website hosting is a fundamental service for any online platform, including applications such as MarketMateFX. It involves renting space on a server where a website's files and data are stored, making them accessible via the web to users all around the world. The choice of hosting can have profound implications for a website's performance, security, and reliability (14).

Critical aspects to consider when selecting a hosting service for a financial trading platform include uptime guarantees, as any amount of downtime can translate to significant opportunities lost for users. Additionally, hosting services must provide robust security measures to protect sensitive financial data against cyber threats. Load balancing capabilities are also essential to manage traffic spikes effectively, which are common in trading environments due to market volatility.

Scalability is another key consideration. As the user base grows, the hosting service must be able to accommodate this growth without service degradation. Moreover, hosting services should offer regular backups, technical support, and compliance with data protection regulations.

The type of hosting: shared, dedicated, cloud-based, or managed hosting also plays a crucial role in determining the level of control, resource allocation, and cost-efficiency (15). For MarketMateFX, a service that requires high reliability and swift data processing, a dedicated or cloud-based hosting solution with a global content delivery network (CDN) might be the most suitable to ensure fast and secure access to the platform's services, regardless of the user's location.

When searching for a hosting provider for a trading journal platform such as MarketMateFX, it's very important to choose services that can ensure high reliability, exceptional performance, and robust security. Three options for hosting that can meet these demands are Hostinger, IONOS, and GreenGeeks.

Hostinger

Hostinger is known for providing good value with feature-packed plans and a strong emphasis on performance and support.

Advantages:

- **Cost-Effective Plans:** Hostinger's plans are competitively priced, making it accessible for small businesses and personal site owners (16).
- **Comprehensive Features:** Offers a range of features including 100 GB SSD storage, unlimited bandwidth, and free SSL certificates.

Disadvantages:

- **Long-Term Commitment for Lowest Price:** To secure the lowest prices, you need to commit to a long-term plan of up to three years.
- **Renewal Price Spike:** Initial low costs can significantly increase upon plan renewal.

IONOS

IONOS offers steep discounts for new subscribers, which can be very attractive for new businesses starting online (17).

Advantages:

- **Affordable Entry-Level Plans:** Provides a low-cost introduction with services suitable for smaller websites or beginner projects.
- **Free Features Included:** Many plans come with a free domain for the first year and a free SSL certificate.

Disadvantages:

- **Price Increase Upon Renewal:** While initial costs are low, the renewal rates can be higher, impacting long-term budgeting.
- **Complexity in Scalability:** While great for entry-level, scaling up services with IONOS might get complex and pricier.

GreenGeeks

GreenGeeks emphasizes environmental sustainability (18), pairing eco-friendly hosting solutions with robust support.

Advantages:

- Eco-Friendly Hosting: Runs on renewable energy, appealing to environmentally conscious users.
- Solid Performance and Support: Known for excellent customer support and reliable hosting performance.

Disadvantages:

- Higher Cost Compared to Other Cheap Hosts: While still affordable, it's generally slightly more expensive than the lowest-priced hosting services.
- Limited Cheaper Plan Options: The range of features in the lower-priced tiers may be narrower compared to other providers.

Each of these hosts provides a distinct set of advantages that cater to different needs, whether it's cost efficiency, environmental sustainability, or beginner-friendly setups.

Final Choice: GreenGeeks

Choosing GreenGeeks as the hosting provider for MarketMateFX was driven by several key considerations that align well with the trading website's needs and values. GreenGeeks is well known for its reliable hosting performance and excellent customer support (19) which is particularly important in this scenario as there is no prior experience in website hosting present during the development of MarketMateFX.

- Strong Performance: Known for its solid uptime and speed, GreenGeeks ensures that MarketMateFX can offer its users a seamless and efficient online experience.
- Comprehensive Customer Support: GreenGeeks provides robust customer service, which is crucial for resolving any potential issues quickly and maintaining high availability and reliability.
- Cost Consideration: GreenGeeks provides relatively cheap website hosting options especially for newer users. (I paid 12.60 for 3 months of hosting).
- Licensing: GreenGeeks provide great support in ensuring that all licensing is included for open source libraries before hosting.

By choosing GreenGeeks, MarketMateFX now has a lifeline that encourages more users to start joining as website hosting indicates reliability and longevity.

Software Licensing

Licensing is a fundamental aspect of software development and distribution, serving as a legal framework that defines how software can be used, shared, modified, and distributed by others. Licensing is very important especially when website hosting comes into account for a trading website such as MarketMateFX.

- **Protection of Intellectual Property:** Licensing establishes legal protection for the creators, ensuring they retain ownership and control over their creations. It allows them to dictate how their software is used and under what conditions, helping to prevent unauthorized use or redistribution (20).

Licensing is a critical component of the software ecosystem. It not only protects the rights of creators but also clearly outlines the rights and responsibilities of users, contributing to a healthy, legal, and innovative software environment.

As React and Node.js are used for MarketMateFX, licensing terms need to be followed correctly to avoid legal issues. Node.js is licensed under the MIT License, which is one of the most permissive and straightforward licenses in the open-source community.

Key Aspects of the MIT License for React and Node.js:

- **Minimal Restrictions:** The MIT License allows users to use, copy, modify, merge, publish, distribute, sublicense, and sell copies of the software freely. This flexibility is particularly beneficial for both commercial and open-source projects, as it imposes very few restrictions on how the software can be used (21).
- **Requirements:** The primary requirement of the MIT License is that you must include the original copyright notice and the license text in any substantial copies or distributions of the software. This typically means including the MIT License text in the Node.js files used in your project or in a location within your documentation.

Implementation for MarketMateFX:

For MarketMateFX, complying with the React.js and Node.js licensing terms means that the MIT License text needs to be included in the project documentation. It is common practice to place this in a `LICENSE` file in the root directory of the project repository (22).

By including the required MIT License text and acknowledging Node.js appropriately, MarketMateFX can freely use React and Node.js without concerns about licensing fees or complex legal procedures. This ease of use shows one of the many advantages of choosing open-source tools like React and Node.js for developing scalable and efficient web applications such as MarketMateFX.

This is also another advantage that contributed to the choice of GreenGeeks for website hosting as they offer exceptional customer service that extends beyond typical hosting support, assisting with technical aspects like code compression and proper inclusion of software licenses. They are adept at helping clients optimize their web applications for performance, which includes compressing JavaScript, CSS, and other assets to improve load times and efficiency. GreenGeeks support is very knowledgeable about open-source licensing requirements and can guide users on how to correctly include license references in their projects. This was especially useful for the successful website hosting of MarketMateFX.

```
* @license React
* use-sync-external-store-with-selector.production.min.js
*
* Copyright (c) Facebook, Inc. and its affiliates.
*
* This source code is licensed under the MIT license found in the
* LICENSE file in the root directory of this source tree.
```

Figure.5 - Example of MIT license

Apache Licence

The Apache License is similar to the MIT licence. It is a free software license written by the Apache Software Foundation (ASF) (23). It allows users to freely use, modify, distribute, and sell both original and derivative works, with no requirement for modifications to be open-sourced. Once again this is beneficial as the use of Node.js and React open source libraries are used.

Key Features of the Apache License:

- Similarly to the MIT License, the Apache License is permissive and allows the use of licensed software in free and proprietary projects. Software under this license can be integrated into applications without the obligation to release the source code of the components.
- Similarly to MIT, Users must include a copy of the license within their code.

Versions:

The Apache License 2.0, released in 2004 (24) , is the most current version and has gained significant popularity due to its clear terms and compatibility with other major open-source licenses.

The Apache License is favored in many open-source community settings because it provides a legal protection framework without compromising the developers' flexibility to use the software as they see fit. This makes it an excellent choice for projects that wish to be used widely in both open-source and commercial environments.

GreenGeek Customer Support also contributes to ensuring that this license is included within the code before website hosting.

2.6. Conclusions

In the development of MarketMateFX, a range of modern technologies has been chosen to ensure that the platform is robust, scalable, and responsive. Each technology choice is purposefully selected to contribute to an efficient and user-friendly trading journal application.

- Database - Firebase
- Front-end - ReactJS
- Back-end - NodeJS
- Web Application - JavaScript
- Website Hosting - GreenGeek

Using these technologies, MarketMateFX is built to offer a seamless, efficient, and secure experience to its users. Each component plays a crucial role in ensuring the platform not only meets the current demands of its users but is also ready for future enhancements and scalability. This technological foundation supports MarketMateFX's objective to be a leading tool in the Forex trading community, facilitating comprehensive trade analysis and effective decision-making.

3. Experiment / Software Design

3.1 Introduction

This section of this report lays down the foundation for understanding the architectural framework and methodologies used in the development of MarketMateFX. This part documents the structured approach taken to ensure that the platform not only meets the specified requirements but also provides a robust, scalable, and maintainable system in the long run.

In software design, the focus is on defining and organizing the system components, their interfaces, and the data that flows between them. The design decisions covered in this section address how MarketMateFX integrates diverse technologies such as Firebase, ReactJS, NodeJS, and JavaScript to create a cohesive and efficient trading journal application. This integration is critical to achieving a seamless user experience and reliable backend services.

By detailing the software architecture, component diagrams, and interaction flows, this section aims to provide stakeholders with a clear view of how MarketMateFX is structured. It also serves as a guide for future development and maintenance efforts, ensuring that the platform can evolve in response to new business requirements or technological advancements. This proactive approach to software design is crucial for maintaining the competitiveness and relevance of MarketMateFX in the fast-paced world of Forex trading.

3.2. Software Methodology

3.1. Research of Methodologies:

3.1.1 Agile Scrum:

Scrum is a lightweight software development methodology that focuses on having small time-boxed "sprints" of new functionality that are incorporated into an integrated product baseline (25).

It is a dynamic approach to project management and application development that prioritises collaboration, adaptability, and continuous feedback. It is perfectly suitable for projects like MarketMateFX, which plans to operate in a fast-paced and constantly changing environment.

In Agile Scrum, the project is divided into time-bound iterations known as "sprints," lasting 2-4 weeks. Each sprint focuses on delivering a specific set of features or functionality of the application. The process begins with a product backlog, a prioritised list of all project requirements/tasks. The product owner, responsible for defining and managing this list, works closely with the development team and a Scrum leader.

During sprint planning, the team selects items from the product backlog to work on during the upcoming sprint. Clear goals and definitions of "done" are established for each item. Daily standup meetings are held to ensure everyone is aligned, and any challenges are addressed promptly.

Advantages of Scrum:

- Flexibility - Agile Scrum allows for changes and adjustments to be made at the end of each sprint. This adaptability is crucial in dynamic environments like the financial markets.
- Rapid Delivery - By breaking the project into manageable sprints, Agile Scrum enables the delivery of working increments of the product at the end of each sprint. This means that valuable features can be delivered sooner.

Disadvantages of Scrum:

- Complexity - Scrum may seem complex to teams who are new to the methodology. It requires dedicated roles, such as Scrum leader/master and product owner, and good following to Scrum practices. This also makes it unsuitable for use on solo projects.

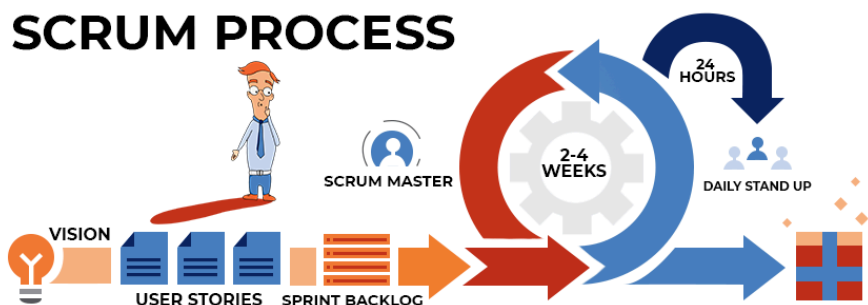


Figure.6 - Scrum Process

3.1.2 Waterfall Method:

The Waterfall methodology is a traditional and linear approach (26) to project management and application development. It is characterised by a sequential flow of phases, with each phase building upon the deliverables of the previous one.

How Waterfall Works?:

- 1. Requirements:** The project begins with a detailed documentation of requirements. This phase involves gathering and specifying all project requirements, often in the form of a detailed project scope.
- 2. Design:** Once the requirements are established, the design phase commences. Designers and architects create a detailed blueprint for the project, outlining the system's structure, user interfaces, and technical specifications.

3. Implementation: With the design in place, developers can start building the project according to the specified requirements and design specifications. This phase involves writing code, developing features, and creating the software or product.

4. Testing: After development, the project moves into the testing phase. Testers evaluate the product's functionality, performance, and quality. Any issues or defects found are documented and addressed.

5. Deployment: Once testing is complete and all issues are resolved, the project is deployed or released to the end-users/customers.

6. Maintenance: The final phase involves maintenance and support. Updates, bug fixes, and improvements are implemented as needed to ensure the product's continued functionality and performance.

Advantages of Waterfall:

- **Structured Approach** - Waterfall provides a clear and structured framework for project development. Each phase has well-defined deliverables and objectives.
- **Ease of Management** - The linear nature of Waterfall makes it relatively easy to manage and track project progress.

Disadvantages of Waterfall:

- **Limited Flexibility** - Waterfall is inflexible when it comes to accommodating changes in requirements. Any alterations usually require going back to the initial phases.
- **Long Delivery Time** - Waterfall projects tend to have longer delivery times, as each phase must be completed before the next one starts.

The Waterfall Method

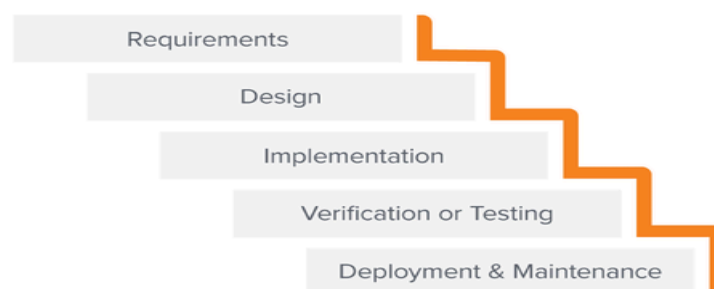


Figure.7 - Waterfall Method

3.1.3. Waterfall Method vs Agile Scrum:

The decision to adopt the Waterfall methodology for this specific project is based on its compatibility with the project's characteristics and requirements. Waterfall's structured and sequential approach aligns well with the needs of this project, providing a clear and well-defined path for development.

Waterfall's emphasis on planning and documentation upfront ensures that all project requirements are thoroughly understood and documented, reducing the likelihood of changes during development. While Scrum offers flexibility and adaptability, the stability and predictability offered by Waterfall make it a suitable choice for a project like MarketMateFX, where precision, regulatory compliance, and well-defined requirements are paramount. Waterfall methodology is also a lot more comfortable to implement and follow as a solo developer.

3.3. Requirements Gathering

ERD Diagram

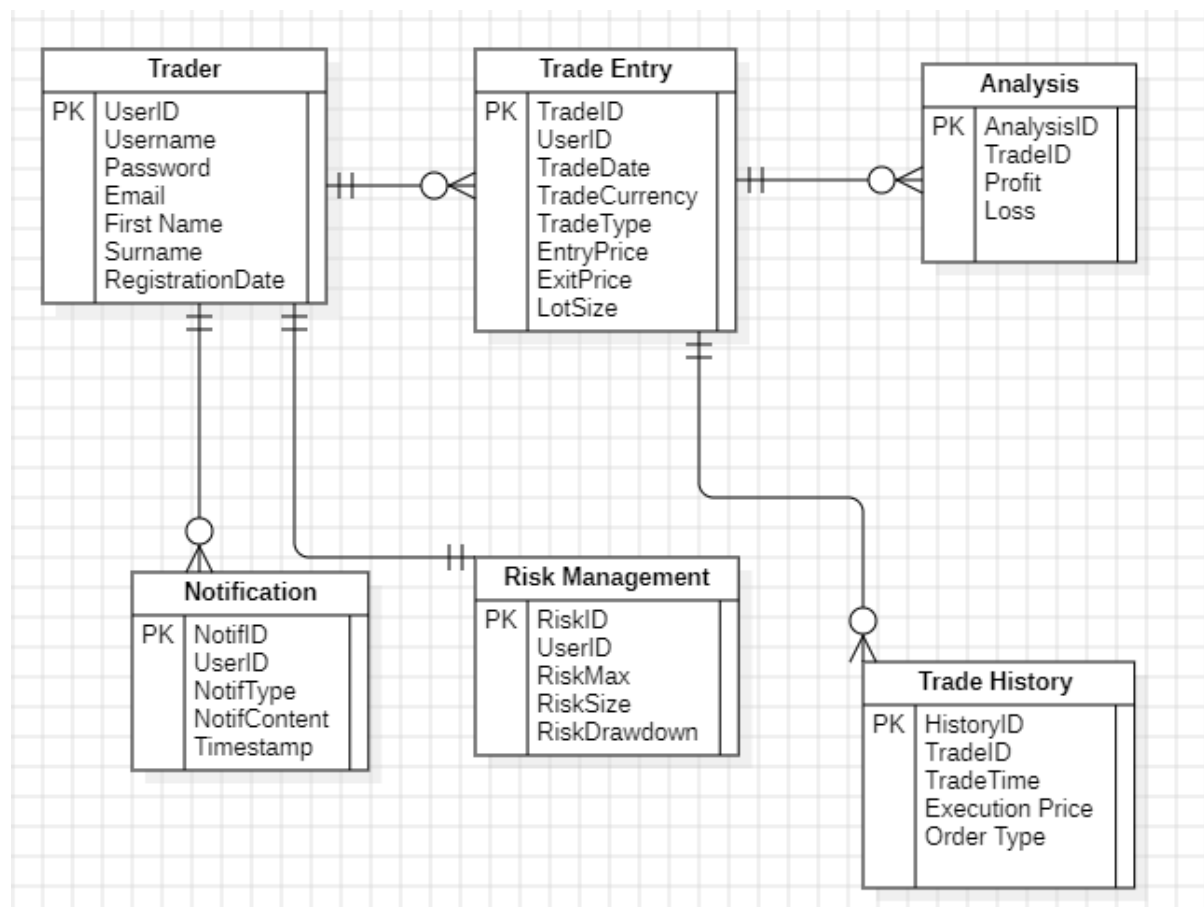


Figure.8 - ERD

This Entity-Relationship Diagram (ERD) represents the data model of MarketMateFX. It shows how the different entities (data objects) relate to each other within the system.

Entities and Attributes:

Trader:

Represents the users of the platform.

Attributes:

- UserID (PK): The primary key uniquely identifying each user.
- Username: The trader's chosen username for the platform.
- Password: Authentication credential for the user.
- Email: Contact email address of the user.
- First Name: Users first name.
- Surname: Users last name.
- RegistrationDate: The date when the user registered on the website.

Trade Entry:

Records the details of a trade made by a trader.

Attributes:

- TradeID (PK): Unique identifier for the trade entry.
- UserID (FK): Foreign key that links to the Trader entity.
- TradeDate: The date the trade was executed.
- TradeCurrency: The currency pair that was traded.
- TradeType: The type of trade (Buy or Sell).
- EntryPrice: The price at which the trade was entered.
- ExitPrice: The price at which the trade was closed.
- LotSize: The size of the trade.

Analysis:

Contains the results of an analysis on a trade.

Attributes:

- AnalysisID (PK): Unique identifier for the analysis record.
- TradeID (FK): Links to the Trade Entry entity.
- Profit: The profit amount from the trade.
- Loss: The loss amount from the trade.

Notification:

Stores information about notifications sent to traders.

Attributes:

- NotifID (PK): Unique identifier for the notification.
- UserID (FK): Associates the notification with a specific trader.
- NotifType: The type of notification.
- NotifContent: The contents or message of the notification.
- Timestamp: The time the notification was sent.

Risk Management:

Holds data related to the risk management parameters set by the trader.

Attributes:

- RiskID (PK): Unique identifier for the risk record.
- UserID (FK): Links to the Trader entity.
- RiskMax: Maximum risk allowed by the user.
- RiskSize: The size of risk the trader is willing to take per trade.
- RiskDrawdown: Maximum drawdown level set by the user.

Trade History:

Archives the history of trades.

Attributes:

- HistoryID (PK): Unique identifier for the historical trade record.
- TradeID (FK): Connects to the Trade Entry entity.
- TradeTime: The time at which the trade was executed.
- ExecutionPrice: The price at which the trade was executed.
- Order Type: The type of order used to execute the trade, e.g., market order, limit order.

Relationships:

- Trader to Trade Entry: A one-to-many relationship indicating that one trader can have multiple trade entries.
- Trader to Notification: A one-to-many relationship showing that one trader can receive multiple notifications.
- Trader to Risk Management: A one-to-one relationship suggesting that each trader has one set of risk management parameters.
- Trade Entry to Analysis: A one-to-one relationship meaning each trade entry has a corresponding analysis.
- Trade Entry to Trade History: A one-to-many relationship indicating that each trade entry can have multiple historical records.

Symbols:

- PK: Primary Key - A unique identifier for each record in an entity.
- FK: Foreign Key - An attribute that creates a link between two entities.

This ERD encapsulates the structural model of the data, which is vital for creating and maintaining the database that supports the functionalities of MarketMateFX. It ensures that the system is built on a strong foundation, with clear definitions of how data elements interact and are accessed within the platform.

Use-Case Diagram

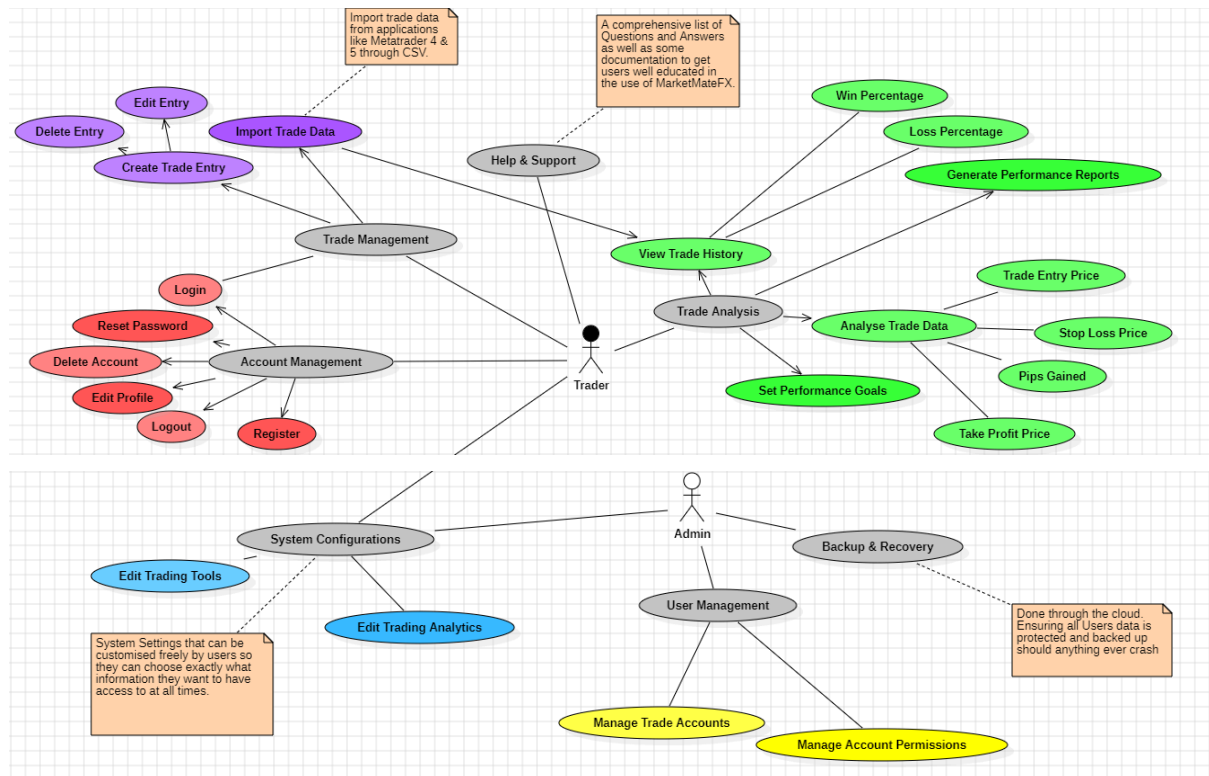


Figure.9 - Case Diagram

This use case diagram above is a visual representation of the interactions between users and the system for MarketMateFX. The diagram categorizes the functional requirements of the system by showing the various actions (use cases) that different types of users can perform.

Actors:

- **Trader (User):** The primary user of the system, engaging in actions related to trading on the website.
- **Admin (Developer):** A user with administrative privileges who manages the system's configurations and user accounts.

Trader Use Cases:

- **Data Entry:** This includes creating new trade entries and editing existing ones.
- **Trade Management:** Allows the trader to import trade data, view trade history, and perform trade analysis.
- **Account Management:** Traders can log in, reset passwords, delete accounts, edit profiles, and log out. There's also a registration function for new users.
- **Help & Support:** Functionality for traders to access help resources or contact support.
- **Performance Tracking:** Traders can set performance goals and generate performance reports. This is further broken down into more specific actions like analyzing trade data, reviewing win/loss percentages, and tracking other metrics like pips gained or trade profit/loss.

Admin Use Cases:

- System Configurations: Allows the admin to edit trading tools and trading analytics settings, ensuring that the system is configured to match the trading environment requirements.
- User Management: Involves managing trade accounts, managing account permissions, and the broader system backup and recovery. This is crucial for maintaining the integrity and security of the system.

The use cases are connected to their respective actors with lines, showing who can perform each function. This kind of diagram is essential for the system development life cycle because it helps ensure that all necessary functionalities are understood and implemented according to user needs. This is also a valuable information tool as all stakeholders can clearly see what is planned for the overall system.

Class Diagram

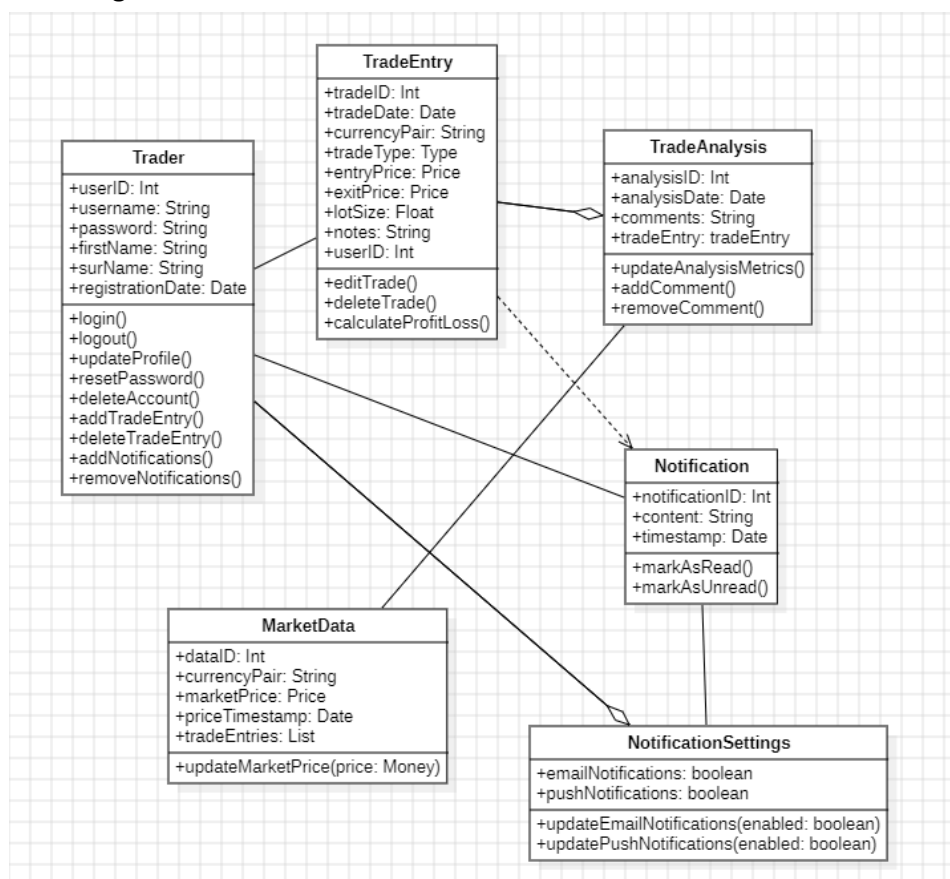


Figure.10 - Class Diagram

The above class diagram represents the structure of MarketMateFX, detailing the classes, their attributes, methods, and the relationships between them.

Class: Trader

- **Attributes:** Defines the properties of a trader, including identification, authentication, personal information, and registration details.
- **Methods:** Showcase actions a trader can perform, such as logging in, updating profile, resetting password, and managing trade entries and notifications.

Class: TradeEntry

- **Attributes:** Holds all the details of a trade transaction, including identifiers, trade-specific data like date, type, prices, size, and the trader associated with it.
- **Methods:** Include actions to edit, delete, and calculate profit or loss from a trade entry.

Class: TradeAnalysis

- **Attributes:** Holds the information regarding the analysis of trades, containing identifiers, the associated trade, and metrics for evaluating the trade.
- **Methods:** Comprise functions to update an entry, add or remove comments, and add or retrieve analytical metrics.

Class: Notification

- **Attributes:** Defines the structure of notifications sent to traders, with identifiers, content, and timestamps.
- **Methods:** Include the functionality to mark a notification as read or unread.

Class: MarketData

- **Attributes:** Holds information about market prices for currency pairs at specific times.
- **Methods:** Includes a method to update market prices, reflecting the dynamic nature of the financial markets.

Class: NotificationSettings

- **Attributes:** Specifies the trader's preferences for receiving notifications, including through email or push notifications.
- **Methods:** Holds the functions to enable or disable these notifications based on the user's preferences.

Relationships:

- Associations between classes are shown, indicating how they interact. For example, the Trader class is associated with the TradeEntry class. This shows that traders have one or more trade entries.

This class diagram serves as a foundational blueprint for the system design and guides the programming structure of MarketMateFX that is expected. It is detrimental to the development phase, ensuring that each component of the system is well-defined and that the interactions between components are clearly understood.

3.4. Overview of System

The goal of MarketMateFX is to give traders a smooth and effective platform for monitoring, evaluating, and improving all trading activity. The operations of the system comprise a number of essential elements that come together to produce a thorough trading journal experience.

Data management is a fundamental component of MarketMateFX's operations. The system performs exceptionally well at handling trading data, providing a safe haven for crucial data such as trade entries and exits, currency pairings, and trading tactics. Users may simply access and securely save this trade data, which guarantees accurate record-keeping.

MarketMateFX places a high focus on security, and to safeguard user accounts and trading information, strong user authentication is implemented. These authentications improve the overall security and user experience of the system by protecting sensitive data and facilitating a personalised user experience.

MarketMateFX users have an easy-to-use interface when entering their trade data. The procedure is made simpler by the system, which makes it simple for traders to record exact entry and exit points, trade lengths, and strategy specifics. Users can concentrate on trading thanks to this efficient data entry method, which saves time and effort.

With the help of this program, traders can use a robust analytical tool that offers comprehensive insights into their trading performance. Users are able to assess the performance of their tactics, pinpoint their advantages and disadvantages, and make data-driven choices about their upcoming trades. In order to maximise trading tactics and overall trading performance, these analytical skills are essential.

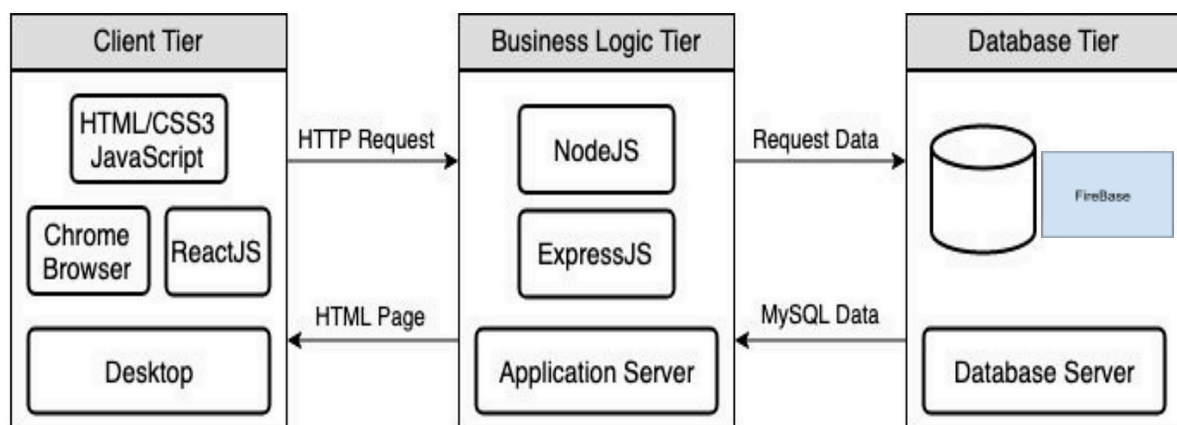


Figure.11 - System Architecture

The diagram above showcases the three-tier system architecture model that is followed for MarketMateFX. The diagram outlines how the client, business logic, and database tiers interact with one another.

Client Tier:

Consists of HTML/CSS3 and JavaScript, which are the foundational technologies for creating web pages. They run in a web browser, perfect for MarketMateFX.

The client tier is where the user interacts with the application. Here, ReactJS is used to build the user interface components. The desktop represents the user's hardware from which they access the web browser.

- **Data Flow:** The client tier sends HTTP requests to the server and receives HTML pages in response, which are then rendered by the browser.

Business Logic Tier:

Contains Node.js as the runtime environment.

The business logic tier processes HTTP requests from the client tier, interacts with the database, and implements the application's business logic. Node.js facilitates the running of JavaScript on the server-side.

- **Data Flow:** The business logic tier receives data from the client tier, processes it and sends back the appropriate responses. This tier runs on an application server.

Database Tier:

Utilizes Firebase, a NoSQL database management system.

The database tier is responsible for storing and managing all the data in the application, such as user data, transaction records, etc. The database server is where data is stored, retrieved, and manipulated.

- **Data Flow:** The business logic tier communicates with the database server, sending requests for data (Request Data) and receiving the required information to process business logic or respond to client requests.

The arrows indicate the direction of the data flow. Requests flow from the client tier to the business logic tier and, if needed, to the database tier. Responses and requested data flow back from the database to the business logic tier and then to the client tier.

The three-tier architecture model presented in the diagram above is particularly advantageous for MarketMateFX for several reasons. By segregating the system into the client, business logic, and database tiers, MarketMateFX benefits from a modular design that enhances maintainability and

scalability. This separation of concerns allows the platform to handle complex data processing and analysis specific to trading, such as managing trade entries and executing risk analysis algorithms, within the business logic tier without affecting the client interface.


3.X. Wireframe

Wireframes play a huge role in the requirements gathering process as it acts as a visual guide that represents the framework of a website. They are essentially the blueprints that illustrate the layout of various pages and the placement of elements such as navigation menus, content areas, and interactive features. During requirements gathering, wireframes help stakeholders, including developers, designers, project managers, and clients, envision the program's structure and discuss its functionality without the distraction of design elements like color schemes or graphic styles.

<https://www.experienceux.co.uk/faqs/what-is-wireframing/#:~:text=A%20wireframe%20is%20commonly%20used,design%20and%20content%20is%20added>.

Wireframes can also reveal the relationships between different pages and how users will navigate through the system, providing insights into the workflow and overall user journey. This helps in defining the system's information architecture and the interactive aspects of the interface, which are critical components of the system requirements. By incorporating wireframes into the requirements gathering process, the necessary functions can be documented and contribute to successful system development.

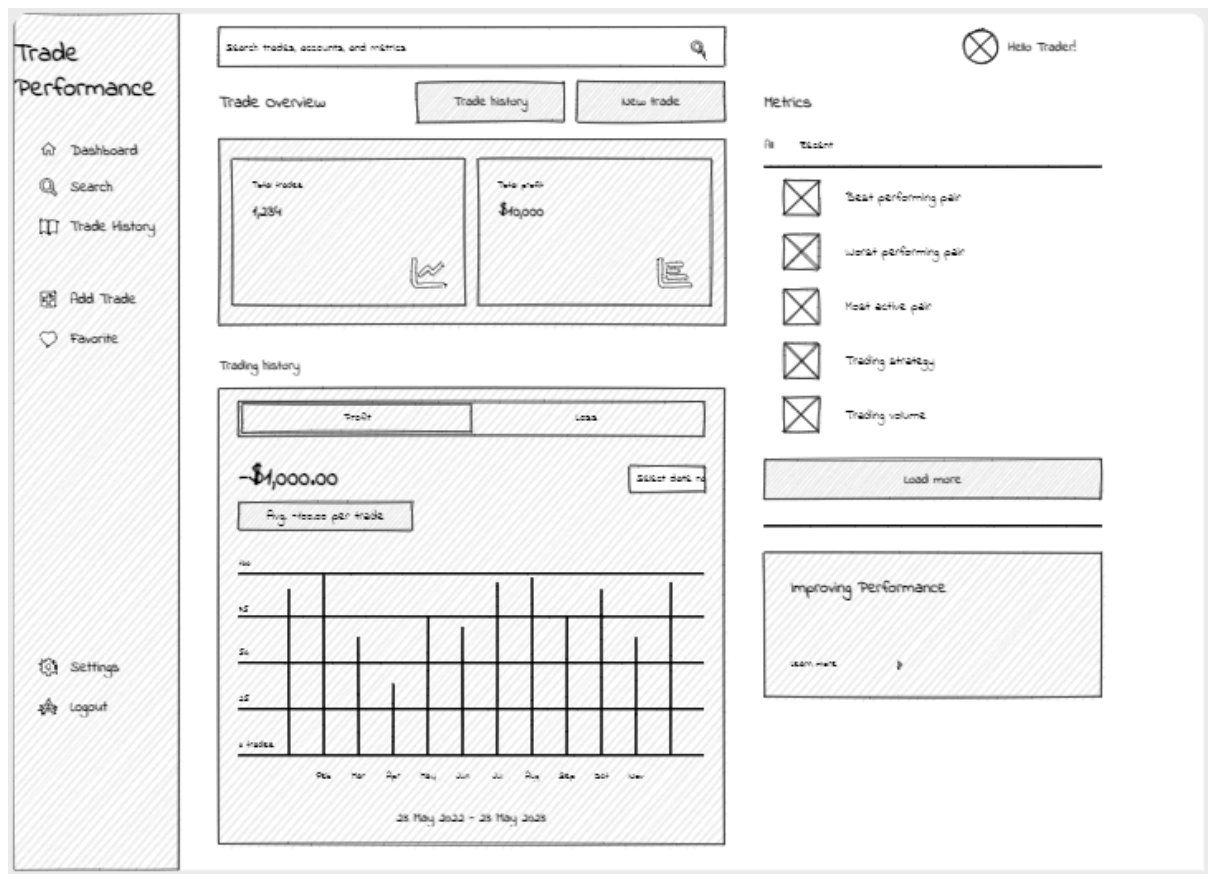
Below are the wireframes that were developed during the initial requirements gathering of MarketMateFX:



The wireframe shows a registration form for MarketMateFX. At the top, the logo 'MarketMateFX' is centered, with a small icon of a line graph below it. Below the logo, there are three input fields: 'Full Name' with the placeholder 'Enter your full name', 'Username' with the placeholder 'Create a username', and 'Location' with the placeholder 'Choose your location'. At the bottom, there are two buttons: 'Previous' and 'Continue'.

Figure.12 - Login Wireframe

The wireframe above shows the primary design for the user registration page for MarketMateFX. At the top, the main logo for the website sits prominently, to reinforce website recognition. Below the logo, there are fields for user input, which are integral to the sign-up process: one for the full name of the user, a second for choosing a username, and a third for selecting a location. At the bottom of the wireframe, there are two buttons 'Previous' and 'Continue' which indicates a multi-step registration process, allowing users to navigate forward and backward as they provide their information. This wireframe serves as a strategic tool in the development process, mapping out the essential elements of the registration interface and providing a visual representation of what to follow during development.



The above wireframe illustrates the user interface for the "Trade Performance" section of the MarketMateFX platform. This will act as the home screen dashboard. It is designed with a clean, simple and functional layout to provide users with insights into their trading activities.

On the left, a navigation bar offers quick access to different sections like Dashboard, Search, Trade History, Add Trade, Favorite, Settings, and Logout. The top features a search bar, to look up historical trades and relevant data

The main area is divided into two sections. On the left, under "Trade Overview," there are components displaying key performance indicators such as win ratio and net profit, presented in both numerical and graphical formats for quick, at-a-glance consumption of information. Below this, there is a detailed "Trading History" graph charting profits and losses over time, allowing traders to visually track their performance.

On the right side, there's a "Metrics" panel with checkboxes, allowing users to select which data points they want to view or compare, such as best/worst performing pair, win/loss percentage, and trading volume. Below the metrics, there's a section for "Improving Performance" with text input, which will be used to set new goals or take notes on trading strategy adjustments.

Overall, this wireframe is structured to focus on usability and clarity of information, providing users with a comprehensive view of their performance and easy navigation to deeper analytics for enhancing their trading strategies.

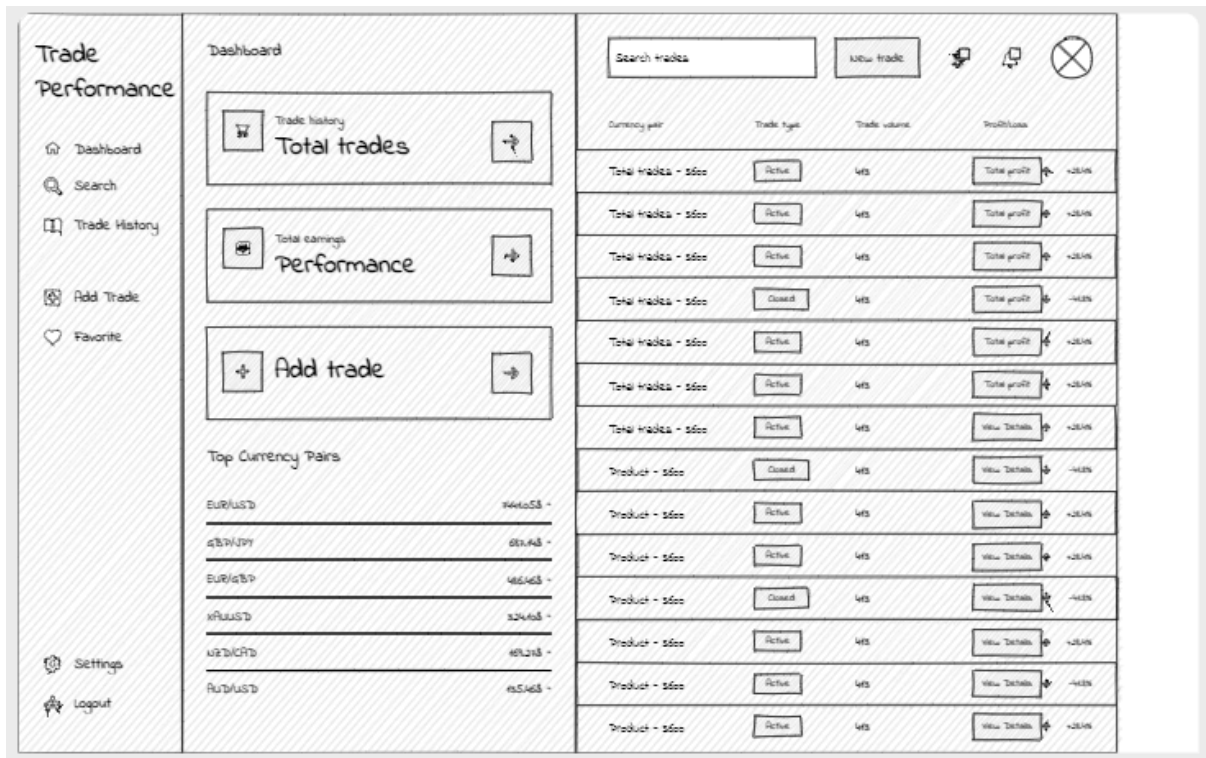


Figure.13 - Dashboard Wireframe

The third wireframe above presents a user interface for the "History" section of MarketMateFX, focusing on trade performance and management. This layout is specifically organized to give traders a comprehensive overview of their activities and performance metrics.

The central area of this page is divided into several components. At the top, there is a "Trade History" section summarizing "Total trades" and a "Performance" module showing "Total earnings," both providing quick insight into the trader's recent activities and success rate. Below these summaries is a big "Add Trade" button, allowing for users to input new trading data.

Below the action buttons, there is a list detailing the "Top Currency Pairs," ranked by performance metrics, to inform traders about their most profitable trades and to identify market trends.

On the right side of the wireframe, there is a detailed "Search" component with filtering options to refine search results by parameters like currency pair, trade type, win/loss status, and more. The search results are displayed in a tabular format listing individual trades with options to edit or delete, indicating functionality for managing and reviewing specific trade entries.

Add a Trade

Dashboard
Search
Trade History
Add Trade
Favorite
Settings
Logout

Search for a currency pair

Add Trade

Currency Pair & Description

Currency Pair Name

Trade Description

Enter trade description

Trade Strategy

Enter trade strategy

Enter trade risk %

Enter trade profit

Enter trade loss

Trade Images & CTR

Trade Image

Upload Trade Chart

Call to Action

Enter call to action

Trade Price

Trade Amount

Enter trade price

Trade Category & Attributes

Trade Category

Select trade category

Trade Type

☒ Long Trade
☒ Short Trade
☒ Swing Trade

☒ Day Trade
☒ Scalping Trade
☒ Position Trade

☒ Trade Strategy
☒ Technical Analysis
☒ Fundamental Analysis

☒ Trade Indicator
☒ Moving Average
☒ Bollinger Bands

☒ RSI
☒ MACD
☒ Stochastic oscillator

Trade Tag

Enter trade tag

Add Tag

Trade Preview

Trade Details

Submit

Publish Trade

Schedule Trade

Save Draft

Trade Schedule

Select Date

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Start Time

End Time

Trade

Get Schedule

Figure.13 - Add Trade Wireframe

The fourth wireframe showcases the design of an "Add a Trade " page within MarketMateFX, focusing on the detailed entry of a new trade. It is structured to facilitate the user input of comprehensive trade-related data, which is crucial for maintaining accurate and actionable trading records.

Overall, these wireframes outline a user-friendly and data-centric approach to journaling trade data, with a big emphasis on the ease of recording and reviewing comprehensive trading details. The

layout promotes meticulous data entry, which is essential for a platform such as MarketMateFX, that relies on detailed analytics and performance tracking to provide value to its users.

3.X. Conclusions

The requirements gathering process has been instrumental in laying the foundation for MarketMateFX' success. Through thorough analysis and documentation, valuable insight has been gained into the needs, preferences, and constraints that will shape the development phase of this particular project. By documenting these requirements, misunderstandings are minimised and the likelihood of delivering a solution that meets or exceeds expectations is high. Moving forward, the identified requirements will serve as the foundation of this website guiding the design, development, and implementation stages, ensuring that the final product not only addresses the stated needs but also enhances user satisfaction and achieves all project goals.

4. Experiment / Software Development

4.1. Introduction

The Experiment/Software development phase signifies a critical milestone in the project's progression, transitioning from requirements gathering to the practical implementation of the actual software solution. This section discusses the methodologies, tools, and processes employed in the development of MarketMateFX.

During this phase, the primary objective was to translate identified requirements from the previous section into a functional software solution. This involved outlining technical specifications, establishing a robust software architecture, and refining the software to align with the defined criteria. This specific part of the report highlights the significant decisions and challenges encountered along the way, along with the strategies deployed to address them.

Additionally, this section dives into the methodologies utilized for testing, quality assurance, and version control, underscoring the commitment to delivering a software product of high reliability and usability. By providing a comprehensive overview of the development process, stakeholders gain insight into the evolution of the software and are invited to contribute feedback for further refinement.

This Experiment/Software development phase lays the foundation for following stages, including testing, deployment, and general maintenance. Through a systematic approach, the aim is to deliver a software solution that not only meets but exceeds expectations, ensuring user satisfaction and achieving the project's objectives.

4.2. Software Development

Due to the nature of the website being a financial project with many calculations happening simultaneously, many variables are one letter or two letter variables as they are concise, taking up less space and therefore, making the equations themselves easier to read and execute faster.

Project Components:

The most important feature of MarketMateFX is the ability to enter trade data and store it within the database. This is done with a simple enough form submission process.

```
handleSubmit: async (values, formikBag) => {
  let fileUrl = ""; // This variable will hold the URL of the uploaded file
  if (file) { // Check if a file is selected
    const formData = new FormData(); // Create a new FormData object
    formData.append("file", file); // Append the selected file to the FormData object
    try {
      // Try to upload the file to the server
      fileUrl = (await axios.post("upload.php", formData, {
        headers: {
          "Content-Type": "multipart/form-data",
          Accept: "application/json"
        }
      })).data.fileUrl; // Get the URL of the uploaded file from the server's response
    } catch (error) {
      console.error("Error uploading file:", error); // Log an error if file upload fails
    }
  }
  try {
    // Prepare the form data for submission
    values.startDate = startDate; // Set the start date
    values.endDate = endDate; // Set the end date
    values.tradeType = tradeType; // Set the trade type
    values.tags = tags; // Set the tags
    values.tradeAmount = parseInt(values.tradeAmount); // Convert trade amount to integer
    values.tradeRisk = parseInt(values.tradeRisk); // Convert trade risk to integer
    values.fileUrl = fileUrl; // Set the file URL

    const userId = (firebaseUser == null ? undefined : firebaseUser.uid); // Get the user ID
    const timestamp = new Date().getTime(); // Get the current timestamp
    values.timestamp = moment(timestamp).format("DD-MM-YYYY"); // Format the timestamp
    await axios.post(`https://marketmatefx-web-default-rtdb.firebaseio.com/users/${userId}/tradeDetails.json`, values); // Submit
    formikBag.resetForm(); // Reset the form after successful submission
  } catch (error) {
    console.error("Error submitting the form:", error); // Log an error if form submission fails
  }
}
```

Figure.14 - Code Snippet of trade form

This code represents the form submission process that controls the main feature of MarketMateFX. When the form is submitted, the `handleSubmit` function is triggered. This function first checks if a file has been selected for upload, and if so, it uploads the file to a server endpoint using the Axios library. The Axios library is a JS library that is used for making HTTP requests from the web browser and the Node.js environment.

The form data is then prepared by setting various values such as the start date, end date, trade type, tags, trade amount, and trade risk. The `tradeAmount` and `tradeRisk` values are converted to integers, and the user ID and current timestamp are obtained. The form data, including the uploaded file URL, is then submitted to the Firebase Realtime Database endpoint specific to the user currently logged in which is determined by looking at the user ID of the currently logged in user. After

successful submission, the form is reset, clearing the form fields, and any errors encountered during the submission process are logged. Overall, the code above facilitates the collection and storage of trade details in a structured manner, integrating file upload functionality and interacting with a Firebase database for data persistence. This acts as the main component of the website saving the trade details to be pulled later for various components.

Currency Pair & Description

Currency Pair

Currency pair name

Trade Description

Enter trade description

Trade Strategy

Enter trade strategy

Enter trade risk %

Trade Outcome

Select trade outcome

Trade Profit

0

Trade Loss

0

Trade Images & CTA

Trade Images

Upload Trade Chart

Trade Schedule

Trade Price

Trade Amount

Enter trade price

Enter PIPs

Enter stop loss price

Enter take profit price

Figure.15 - Add Trade Form

As MarketMateFX is a financial application, many calculations are ongoing simultaneously to give users the best analytics in real time. One such example is the average risk percentage being calculated based on the risk percentage used in each individual trade by users.

```
Oe = (Re => {
  if (Re.length === 0) return 0;
  const je = Re.reduce((Ze, Ye) => {
    const et = parseFloat(Ye.tradeRisk);
    return isNaN(et) ? Ze : Ze + et;
  }, 0) / Re.length;
  return isNaN(je) ? 0 : je;
})(t);
```

Figure.16 - Risk Function

This code snippet shown above defines an arrow function that calculates the average trade risk value from an array of trade objects input by the user in the form previous. Initially, it checks if the input array (Re) is empty, returning 0 if so. If the array contains trade objects (Ye), it iterates over each object using the `reduce` method to accumulate the trade risk values (Je). Within this iteration, it parses each trade risk value as a floating-point number, skipping invalid values. After calculating the total trade risk value (Ze), it divides this number by the number of trade objects to obtain the average. If the calculated average is `NaN`, indicating that there were no valid trade risk values, the function returns 0; otherwise, it returns the calculated average. As a result, each individual user will get a risk average unique to them and their trade inputs. In the particular example below, this specific user has an average percentage of 2 across their several trades.



Figure.17 - Risk Component

Another similar equation is used to find the Win/Loss percentage that is unique to the logged in user based on the trades they have input on their account.

```
Ie = (function calculateWinLossPercentage(trades) {
  const wins = trades.filter(trade => trade.tradeProfit !== 0);
  const totalProfit = wins.reduce((sum, trade) => sum + trade.tradeProfit, 0);
  const totalTradeAmount = wins.reduce((sum, trade) => sum + parseFloat(trade.tradeAmount), 0);
  return totalTradeAmount === 0 ? 0 : (totalProfit / totalTradeAmount) * 100;
})(t);
```

Figure.18 - Win/Loss Code

The above function calculates the win/loss percentage for a set of trades provided in the `trades` array. It begins by filtering the `trades` array to extract only the winning trades, where the `tradeProfit` property is not equal to 0. This was input by the user when they are adding a trade. Then, it calculates the total profit from these winning trades by summing up the `tradeProfit` values using the `reduce` method. It computes the total trade amount by summing up the `tradeAmount` values of the winning trades. Finally, it calculates the win/loss percentage by dividing the total profit by the total trade amount and multiplying by 100. If the total trade amount is 0, indicating that there are no trades or the trade amounts are invalid, it returns 0. This function provides a concise and

efficient way to compute the win/loss percentage from a given set of trades, considering both the profits and trade amounts.

A minor problem did present itself during the testing of this function. Due to how it was written and calculated based on profit, the calculation would be incorrect if a user had no lost trades. This can be seen below where this specific user has a win rate of 100% but the calculation says the win rate is much higher.



Figure.19 - Win/Loss Component

Another major component of MarketMateFX is user authentication

```
async updateUser(t, n = !1) {
  if (!this._deleted) {
    t && Tt(this.tenantId === t.tenantId, this, "tenant-id-mismatch"),
    n || await this.beforeStateQueue.runMiddleware(t),
    this.queue(async () => {
      await this.directlySetCurrentUser(t),
      this.notifyAuthListeners()
    })
  }
}

async signOut() {
  return bo(this.app) ? Promise.reject(oa(this)) :
  (await this.beforeStateQueue.runMiddleware(null),
  (this.redirectPersistenceManager || this._popupRedirectResolver) && await this._s
  this._updateCurrentUser(null, !0))
}

setPersistence(t) {
  return bo(this.app) ? Promise.reject(oa(this)) :
  this.queue(async () => {
    await this.assertedPersistence.setPersistence(t)
  })
}
```

Figure.20 - Authentication Code

The code snippet above showcases methods within a class responsible for managing user authentication and state updates. The `updateCurrentUser` method is asynchronous and takes two arguments: `t` (a user object) and `n` (a boolean indicating whether to run middleware before updating the user state). The middleware in this specific program is used to check for authentication and also to ensure persistency is maintained through the several different pages of this website.

It first checks if the current user instance has been deleted. If not, it verifies the tenant ID consistency between the current user and the provided user object. If specified, it runs middleware before updating the user state. Then, it asynchronously updates the current user state and notifies authentication listeners. The `signOut` method handles user sign-out functionality by rejecting the

operation if the app is not initialized properly. It runs middleware to perform any necessary tasks before sign-out, such as clearing session data, and then updates the current user state accordingly. The `setPersistence` method sets the authentication persistence mode for the app. The operation is rejected if the website is not initialized correctly and then sets the persistence mode accordingly. This functionality ensures all users are logged in as needed for access to MarketMateFX.

4.X. Problems that arose during development

The initial challenge was learning React's component-based architecture, JSX syntax, and state management, alongside learning Node.js for server-side logic and Firebase for real-time database and authentication. This was overcome through dedicated time to tutorials, documentation, and online articles. Hands-on practice and seeking help from the community via forums and developer communities were instrumental in solidifying understanding and moving forward during the development of MarketMateFX.

Implementing math equations for the website's functionality, such as financial calculations, posed challenges as they occasionally returned incorrect figures. To address this, rigorous debugging, equation scrutiny, cross-checking with trusted sources, and implementation of unit tests were employed. These measures helped identify and rectify errors.

Different web browsers interpret HTML, CSS, and JavaScript differently, leading to inconsistencies in rendering and functionality. To address this, techniques were employed such as adhering to web standards, using modern CSS techniques with fallbacks for older browsers, and testing early and often helped mitigate compatibility issues.

Slow page load times and “laggy” user interactions can lead to poor user experience and decreased engagement. Performance issues were tackled by optimizing assets (such as images, CSS, and JavaScript), implementing lazy loading for non-essential resources, minimizing HTTP requests, and using browser caching for trade data.

Over time, as the codebase grows, maintaining code quality and readability becomes essential to facilitate collaboration and future development. To target this, professional and modern coding standards are followed. The use of linting tools like ESLint also helped to ensure coding standards were followed and to keep coding security risks to a minimum.

These efforts helped with navigation through the complexities of learning new technologies and overcoming these technical hurdles during the development process. Persistence, adaptability, and a commitment to continuous learning were essential for successfully delivering a robust and functional trading journal website.

4.X. Final Look

Below is a look at the final UI of MarketMateFX and the various screens:

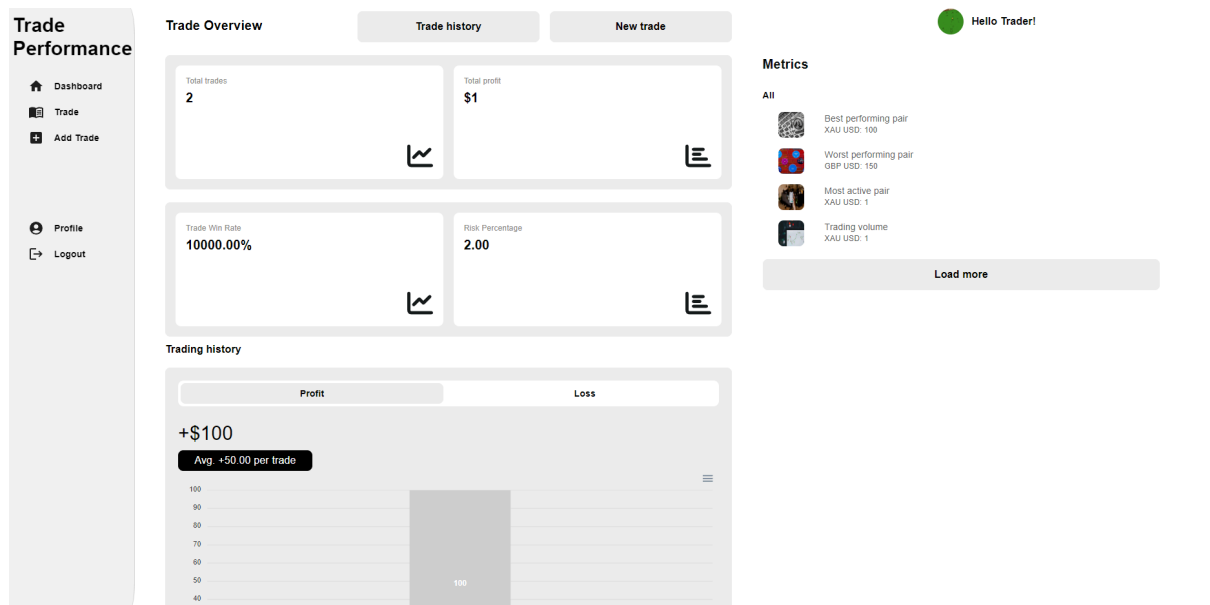


Figure.21 - MarketMateFX Dashboard

The Add a Trade form is designed for users to input trade details and submit them. It includes a sidebar for navigation and a main form area with the following sections:

- Currency Pair & Description:** Fields for Currency Pair, Currency pair name, Trade Description, Trade Strategy, Trade Outcome, Trade Profit, and Trade Loss.
- Trade Images & CTA:** A section for uploading trade images and charts, with an "Upload Trade Chart" button.
- Trade Schedule:** A toggle switch for enabling or disabling the trade schedule.
- Trade Details:** A section for entering trade notes and tags, with an "Add tag" button.

The form also includes a "Submit" button at the bottom for finalizing the trade entry.

Forex

Dashboard

Trade

Add Trade

Profile

Logout

Trader!

Total Trades: 0

Total Pips: 929

Performance Metrics

Trading Strategies

Trading Journal

Technical Analysis

Successful

Trend

Profitable Trades: 1

Trading

Unprofitable Trades: 1

Market

Loss Trade: 93

Risk

Win Trades: 12

Trading

Loss Trades: 24

Trading

Win Trades: 24

Profile

Dashboard

Trade

Add Trade

Profile

Logout

Edit Profile

Email

tester@gmail.com

Username

Test

Full Name

Test

Update Profile

Trade Performance

Dashboard

Trade history

Total trades (2)

Total earnings

Performance (1)

Add trade

Top Currency Pairs

GBP USD

XAU USD

Search trades

New trade

\$

Currency pair	Trade type	Trade volume	Profit/Loss
GBP USD	Short Trade	1	Total Loss
XAU USD	Long Trade	1	Total Profit

Add a Trade

Dashboard

Trade

Add Trade

Profile

Logout

Search for a currency pair

Add trade

Currency Pair & Description

Currency Pair

XAU USD

Trade Description

TESTING

Trade Strategy

Supply And Demand

3

Trade Outcome

Win

Trade Profit

100

Trade Loss

0

Trade Images & CTA


Trade Images

1712843283_66_image_2024-04-11_1447283356.png

Trade Price

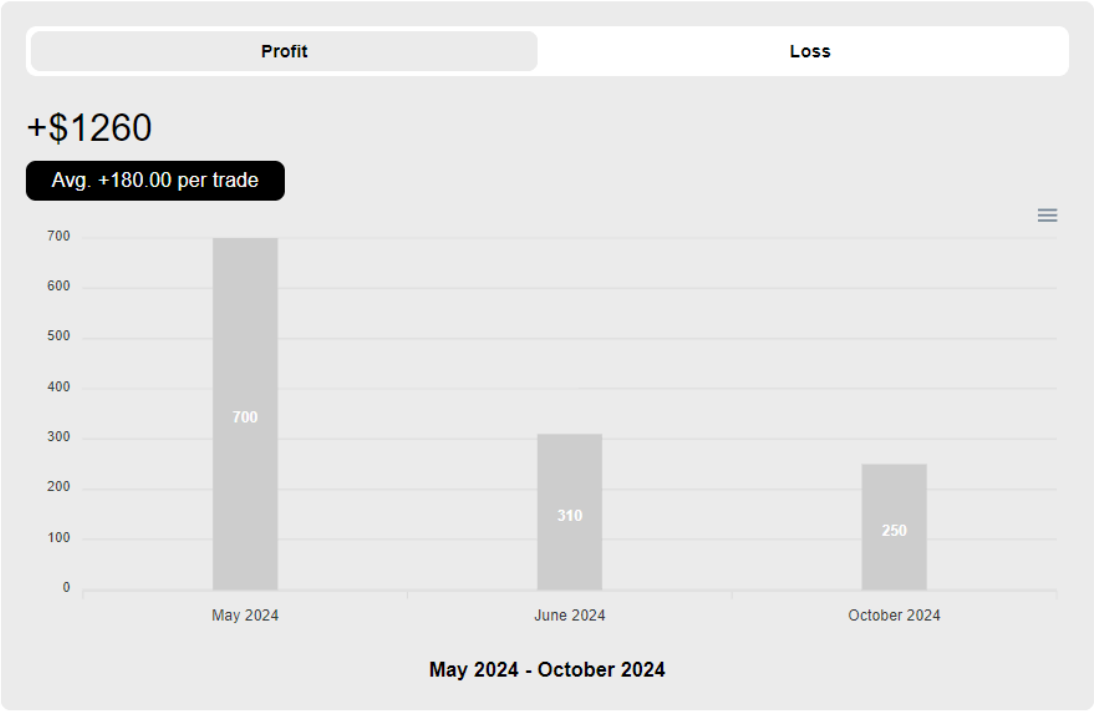
Trade Amount

Trade Preview



Trade Details

Trading history



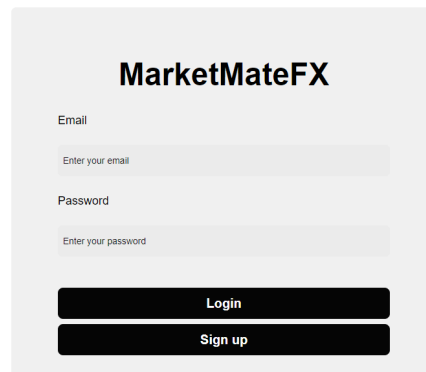
The image shows a login form for MarketMateFX. At the top, the title "MarketMateFX" is centered in a bold, black font. Below the title, there are two input fields. The first is labeled "Email" and contains the placeholder text "Enter your email". The second is labeled "Password" and contains the placeholder text "Enter your password". Below these fields are two buttons: a "Login" button and a "Sign up" button, both in a dark grey color with white text.

Figure.24 - MarketMateFX Login

4.X. Conclusions

The software development section of this report has thoroughly covered the key aspects of building MarketMateFX, including the challenges faced and the solutions implemented. The architecture, security features, and user interface design have been detailed, showing how each part contributes to making the website strong and effective.

Throughout the development process, several issues were tackled such as scalability, security, and system integration, applying practical solutions to ensure that MarketMateFX meets high standards and is ready for future growth

5. Testing and Evaluation

5.1. Introduction

In the development lifecycle of MarketMateFX, testing stands as a crucial phase designed to ensure that the application not only meets its defined functional and technical requirements but also delivers a seamless and robust user experience. Through a series of systematic tests, this phase aims to uncover any defects, performance issues, or usability challenges that could hinder the application's effectiveness. By rigorously examining every component and feature of MarketMateFX, the testing phase guarantees that the final product is not only functional but also reliable and user-friendly, ready to assist all Forex traders in optimizing their trading strategies and achieving their financial goals. This process is essential in building confidence in the application's quality and stability, clearing the way for a very successful launch and a good user adoption.

The main goal of testing is to ensure that the entire basis of this application works as intended for a trading journal. This involves answering some questions such as: Are the correct details being captured? Are statistics being pulled correctly based on input? Can images be saved? Can users have access to their history at any time? Are the objectives of this very project being met?

5.2. System Testing

During the system testing phase of MarketMateFX, lots of attention was directed towards the trade data input functionality. This process involved validating the reliability, accuracy, and user-friendliness of the form, which is absolutely instrumental for traders to document and analyze their Forex trades in a comprehensive way.

The testing protocol included exhaustive scenario-based evaluations, automated stress tests, and thorough manual reviews to see how the system performs under diverse conditions and inputs. Various fields of trade input were investigated by testing users, such as the currency pair selection, strategy description, risk percentage calculation, and the trade price section, including PIPs and P/L figures.

Feedback from the user testing group revealed an area for improvement within the risk percentage input field as this average was not being calculated correctly adding to user confusion. Additionally, the trade image upload feature was found to be lacking in communicative feedback, leaving users uncertain of their upload status.

In response, several enhancements were introduced: a tooltip was integrated into the risk percentage field to provide immediate information upon user interaction, and dynamic calculations were displayed to give real-time feedback as data was entered which could then be seen on the dashboard under "risk". For the image upload process, a visible progress indicator was added alongside a confirmation notification upon successful upload to ensure users received clear and unambiguous feedback.

Post-implementation, these modifications underwent another round of rigorous testing. The result was a significant reduction in user-reported issues, indicating a marked improvement in the trade data input process's effectiveness. The refinements resulted in a user experience that aligned closely with the expectations of Forex traders, as shown by a smooth and efficient data entry process that contributed to the overall utility of MarketMateFX.

The final iteration of the trade data input form, post-rectification and re-evaluation, demonstrated the critical nature of user feedback in the enhancement of system testing processes which presented this form below:

Currency Pair & Description

Currency Pair

Currency pair name

Trade Description

Enter trade description

Trade Strategy

Enter trade strategy

Enter trade risk %

Trade Outcome

Select trade outcome

Trade Profit

0

Trade Loss

0

Trade Images & CTA

Trade Images

Upload Trade Chart

Trade Schedule

Trade Price

Trade Amount


Enter trade price

Enter PIPs

Enter stop loss price

Enter take profit price

Trade Preview



Trade Details

In the testing phase for the metrics section of MarketMateFX, several forms of verification were carried out to ensure that the system accurately pulled and displayed the correct trading information. The metrics are a critical component, as they provide users with insights into their trading performance, including the best and worst performing pairs, the most active pairs, and total trading volume.

The approach to testing these metrics involved:

Verification procedures were established to ensure that the data displayed in the metrics section, such as the best performing pair and worst performing pair, were accurately reflecting the user's trading history. This involved cross-referencing the metrics with the raw data inputs from the added trade logs.

The underlying algorithms that calculate and determine these metrics were tested for accuracy. This included the algorithms that identify the 'most active pair' and compute the 'trading volume'. Each algorithm underwent a white-box testing approach, reviewing the internal processing logic to ensure correctness.

Load testing also ensured that the system could handle a significant volume of data entries and still produce the correct metrics without any performance degradation.





The layout and presentation of the metrics were also reviewed to confirm that users could easily understand and interpret the data presented. This was supported by feedback gathered from usability testing sessions with actual users.

Upon initial testing, discrepancies were identified in the 'Load more' functionality where additional metrics were not loading as expected. To address this, corrective measures were implemented to refine the data fetching and presentation logic. Subsequent retesting confirmed the efficiency of these corrections, with the 'Load more' feature operating seamlessly and additional metrics being accurately displayed without any issues.

The process concluded with a validation phase, where the metrics module consistently demonstrated precise performance, reinforcing the application's reliability and the traders' trust in the system. Through this thorough testing and refinement process, MarketMateFX ensured that its metrics component was robust, accurate, and user-friendly, providing traders with valuable insights into their Forex trading activities which gives this table as shown below:

Metrics

All

	Best performing pair XAU USD: 100
	Worst performing pair GBP USD: 150
	Most active pair XAU USD: 1
	Trading volume XAU USD: 1

Load more

Stress/Load Testing

Stress testing is a critical component of the overall testing strategy for any web-based application such as MarketMateFX. It involves the simulation of a real-world heavy load on the system to validate the application's stability and performance under extreme conditions. This kind of testing is essential to identify the breaking points of an application and to ensure that it can handle high traffic without significant degradation in functionality, which is particularly crucial for trading platforms where real-time performance can have severe implications.

To test MarketMateFX, a custom script was written to perform stress testing specifically on the trade inputs feature of the platform. The JS code below serves as an automated script to simulate the addition of a large number of trade entries to the database in a short period. By executing this script, it can be observed on how the application handles increased data load and concurrent writes. This process helps in finding performance bottlenecks, ensuring that traders can reliably input and analyze their trades even during peak usage times, maintaining a high standard of user experience. This testing also ensures that users with several hundred trade inputs will have the exact same experience as similar users with only two trade inputs.

```
// New trade entry with a unique key
const newTradeRef = tradeRef.push();

// Data for the new trade entry
newTradeRef.set({
  currencyPair: currencyPair,
  tradeDescription: tradeDescription,
  tradeStrategy: tradeStrategy,
  riskPercentage: riskPercentage,
  tradeOutcome: tradeOutcome,
  tradeProfit: tradeProfit,
  tradeLoss: tradeLoss
}, function(error) {
  if (error) {
    // Write failed
    console.error("Failed to write data to Firebase:", error);
  } else {
    // Write successful!
    console.log("Trade data written successfully to Firebase.");
  }
});
}

for(let i = 0; i < 1000; i++) { // Replace 1000 with the desired number of entries for load test
  writeTradeData(
    'EUR/USD',
    'Short EUR/USD due to strong USD data',
    'Short sell on breakout',
    '2%',
    'Profit',
    Math.random() * 250, // Random profit for testing
    Math.random() * -100 // Random loss for testing
  );
}
```

Figure.25 - Test Code

The script above is a simple script used to write 1000 trade entries to the database to ensure MarketMateFX continues to run at a full capacity regardless of the number of trade entries present.

To further stress test the website and ensure that MarketMateFX could withstand the simultaneous creation of user profiles, a small live test was conducted. A group of users was instructed to all create new accounts simultaneously with emails provided. This method of testing was beneficial to identifying how the system manages multiple, simultaneous actions typical of a live trading journal environment.

This stress test ensured that MarketMateFX could handle a surge of new users efficiently without compromising performance or, more critically, the security of individual user data.

The outcome of this hands-on testing provided valuable insights into user experience under peak load conditions and verified that MarketMateFX could deliver a high level of service, ensuring that new users could register and begin using the trading platform without delays or complications.















test7@gmai...		12 Apr ...	1l2w7C29uLXou...
test8@gmai...		12 Apr ...	6vd28KWO3qRL...
test9@gmai...		12 Apr ...	DjE8Ctly3EWBz...
test1@gmai...		12 Apr ...	UgNO4lkMOPZA...
test15@gm...		12 Apr ...	Ujsesb8EgghnR...
test4@gmai...		12 Apr ...	Y4VULUUJXGbz...
test14@gm...		12 Apr ...	Yo2G5nIQAQd7...
test3@gmai...		12 Apr ...	aunCZxlj8paZfw...
test10@gm...		12 Apr ...	fraxDfQsOAM96...
test11@gm...		12 Apr ...	gA3nBBn2vCQkt...
test5@gmai...		12 Apr ...	h9eueby31jRlyX...
test12@gm...		12 Apr ...	jxSmdFY2t0YcA...
test2@gmai...		12 Apr ...	nePwnz0RfoQy...
test13@gm...		12 Apr ...	oVmKk826OaO...
test6@gmai...		12 Apr ...	yXzDcdPB4qb5...

Figure.26 - List of User Profiles

5.3. System Evaluation

Testing evaluation for a website such as MarketMateFX is the process of analyzing the results obtained from various testing activities to determine the quality and readiness of the application.

Below is a full table that was used to evaluate aspects of MarketMateFX to ensure readiness for hosting:

Evaluation Area	Metric	Target	Actual	Status	Notes
Load Performance (Script)	Max Users	100	100	Success	Target number of users were added with no drop in performance
Load Performance (Live Test)	Simultaneous creation of users	20	20	N/A	Test was performed 3 times. No performance drop 2 times. Performance drop 1 time.
Data Segregation	Cross-User Data Leaks	0	0	Success	No data leaks between user profiles.
Trade Input	Trades Added Successfully	1	1	Success	Trade inputs added with no errors.
Trade Input (Stress Test)	Several Trades added Simultaneously.	1000	750	Fail	Small performance drop when hitting 750 inputs.
User Interface	UI Errors	< 1%	<1%	Success	UI presented no significant errors during use.
Security	Unauthorised Access	0	0	Success	No successful unauthorized access attempts during testing.
Features	Implemented features successfully	100%	92%	Fail	Two planned features were not implemented correctly in time for testing.
User Satisfaction	Positive User Feedback	> 80%	>80%	Success	Overall feedback from users was positive.

The evaluation of the system testing for MarketMateFX was instrumental to ensure the platform was not only functional but also robust and user-centric. It provided a strong framework for decision-making regarding the application's launch and ongoing development.

5.4. Conclusions

The system testing and evaluation phase for MarketMateFX has been a rigorous and detailed process, encompassing a variety of testing methodologies designed to ensure the platform operates at the highest standards of performance and security. This approach has highlighted the system's strengths, such as its robust user interface and effective data segregation, which met or exceeded all

target benchmarks. Any areas that did not perform as expected, such as load performance have been clearly identified and documented.

The evaluation has not only confirmed the readiness of many features for launch but also provided invaluable insights that will drive further improvements. It has pointed out the criticality of real-time performance in a trading environment and highlights the importance of maintaining a secure, efficient, and user-friendly platform. With the findings from this evaluation, MarketMateFX can implement targeted optimizations, ensuring that upon launch, the platform will deliver a reliable and secure user experience.

The feedback and data collected from this phase form a strong foundation for continuous improvement in the future.

6. Conclusions and Future Work

Reflecting on the development of MarketMateFX, several lessons emerge about the scope and management of the project. Ambitious goals led to an overly complex project plan that, combined with unforeseen setbacks, placed significant time pressures on development. These challenges occasionally forced the reduction or removal of certain features that, while not critical, would have made the overall functionality much better and contributed to more of an appeal of the application.

If I were to approach this project again, I would consider a more conservative scope from the start. This adjustment would reduce the time pressures and allow for a focused development on core functionalities without the guilt of potentially underperforming. However, the foundation laid by the current version of MarketMateFX is solid, providing strong motivation to continue its development.

Moving forward, I plan to enhance the application by further improving the system and adding even more features to allow for more personalized use of the platform. Implementing functionalities to manage new trading strategies and integrate additional financial instruments will also be crucial as I expand the platform's capabilities.

Overall, I am delighted with the outcome of MarketMateFX.

6.3. Conclusions

With the end of this project, I am delighted with myself to be left with a nice trading journal that I can use myself and I can also let others use. The building of this project definitely contributed to my knowledge of computer science and it was actually enjoyable looking back at it. It was a fun venture to undertake and will definitely benefit me in the future.

Bibliography

1. What is Forex Trading? | Investopedia.com [Internet]. [cited 2023 Dec 5].
Available from: <https://www.investopedia.com/articles/forex/11/why-trade-forex.asp>
2. Understanding the Foreign Exchange Market | Investopedia.com [Internet]. [cited 2023 Dec 5]. Available from:
<https://www.investopedia.com/terms/f/foreign-exchange-markets.asp>
3. Who Controls the Forex Market? | @marketinvestopedia33 [Internet]. [cited 2023 Dec 5].
Available from:
<https://medium.com/@marketinvestopedia33/who-controls-the-forex-market-9f0375fada24>
4. Spot Market and Over-The-Counter | Investopedia.com [Internet]. [cited 2023 Dec 5].
Available from:
[https://www.investopedia.com/terms/s/spotmarket.asp#:~:text=The%20foreign%20exchange%20market%20\(or,or%20a%20future%20price%2Fdate.](https://www.investopedia.com/terms/s/spotmarket.asp#:~:text=The%20foreign%20exchange%20market%20(or,or%20a%20future%20price%2Fdate.)
5. Forward Market: Definition and Foreign Exchange Example | Investopedia.com [Internet]. [cited 2023 Dec 5]. Available from:
<https://www.investopedia.com/terms/f/forwardmarket.asp>
6. Company Overview | trendspider.com [Internet]. [cited 2023 Dec 5]. Available from:
<https://trendspider.com/learning-center/tradervue/>
7. Company Profile | tracxn [Internet]. [cited 2023 Dec 5]. Available from:
https://tracxn.com/d/companies/tradersync/_eXKMW7J_1hf5xmTpZfUhKbCCuS65TzFWNciTx4f7xYQ
8. Features | edgewonk.com [Internet]. [cited 2023 Dec 5]. Available from:
<https://edgewonk.com/features/>
9. Store and Sync Data in Real Time - Firebase | Firebase.google.com [Internet]. [cited 2023 Dec 5]. Available from:
<https://firebase.google.com/products/realtime-database#:~:text=The%20Firebase%20Realtime%20Database%20is,between%20your%20users%20in%20realtime.>
10. What is Mongo DB? [Manual] | mongodb [Internet]. [cited 2023 Dec 5]. Available from:
<https://www.mongodb.com/docs/manual/>
11. What is PostgreSQL? | postgresql.org [Internet]. [cited 2023 Dec 5]. Available from:
<https://www.postgresql.org/about/>
12. Eloquent - Getting Started | laravel10.x [Internet]. [cited 2023 Dec 5]. Available from:
<https://laravel.com/docs/10.x/eloquent>
13. What is JavaScript? [EN-US] | developer.mozilla [Internet]. [cited 2023 Dec 5]. Available from:
https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript
14. What is Webhosting? | wix.com [Internet]. [cited 2024 Mar 24].
Available from:
https://www.wix.com/blog/web-hosting?utm_source=google&utm_medium=cpc&utm_campaign=13774768257^126077909722^search%20-%20dsa&experiment_id=^531699814067^&gad_source=1&gclid=CjwKCAjwT-OwBhBnEiwAgwzrUqETpniEgag9yMyuCB25sKynDtgvBjD5t6KcGabRSN5JRxBxq5flBoC6wEQAvD_BwE
15. Types of web hosting | www.techradar.com [internet]. [cited 2024 Mar 24].
Available from:

<https://www.techradar.com/web-hosting/what-are-the-different-types-of-web-hosting#:~:text=Cloud%20Hosting%3A%20Works%20best%20for,over%20the%20hardware%20and%20software.>

16. Claim Deal Now! | hostinger.com [basket] [Internet]. [cited 2024 Mar 24].
Available from: <https://cart.hostinger.com/pay/cd3b5415-76be-4432-8ecf-ad5ee44caae9>
17. WordPress Hosting | IONOS [Internet]. [cited 2024 Mar 24].
Available from: <https://www.ionos.com>
18. Host a website now! | GreenGeeks.com [Internet]. [cited 2024 Mar 24].
Available from:
https://www.greengeeks.com/web-hosting?utm_source=google&utm_medium=sem&utm_campaign=global&utm_content=brand&gad_source=1&gclid=CjwKCAjwT-OwBhBnEiwAgwzrUi6odCoQfLUeJphcws7ywAef7VAVBTxYMHl8emWZGtmERUzy10ERoC2blQAvD_BwE#section-pricing
19. About US and our Offers | GreenGeeks.com [Internet]. [cited 2024 Mar 24].
Available from:
<https://www.greengeeks.com/about/>
20. What is software licensing? | www.TechTarget.com [Internet]. [cited 2024 Mar 24].
Available from:
<https://www.techtarget.com/searchcio/definition/software-license#:~:text=Software%20licenses%20typically%20provide%20end,the%20software%20can%20be%20used.>
21. What is MIT? | SNYK [Internet]. [cited 2024 Mar 24].
Available from: <https://snyk.io/learn/what-is-mit-license/>
22. What is the MIT License? | Quora.com [Internet]. [cited 2024 Mar 24].
Available from: <https://www.quora.com/What-is-the-MIT-License>
23. APACHE LICENSE | SNYK [Internet]. [cited 2024 Mar 24].
Available from: <https://snyk.io/learn/apache-license/>
24. Apache License2.0 | apache.org [Internet]. [cited 2024 Mar 24].
Available from: <https://www.apache.org/licenses/LICENSE-2.0>
25. What is Agile Scrum Methodology? | Scrum.aspx Inflectra [Internet]. [cited 2023 Dec 5].
Available from:
<https://www.inflectra.com/Solutions/Methodologies/Scrum.aspx#:~:text=Agile%20scrum%20methodology%20is%20a,with%20a%20Potentially%20Shippable%20Product>
26. A Complete Guide - Waterfall Methodology | business.adobe [Internet]. [cited 2023 Dec 5].
Available from:
<https://business.adobe.com/blog/basics/waterfall#:~:text=The%20waterfall%20methodology%20is%20a,detailed%20documentation%2C%20and%20consecutive%20execution.>