

# MINOR PROJECT REPORT ON

## *“StockPulse”*

Submitted in partial fulfilment of the requirement for  
the award of the degree of  
**Bachelor of Computer Applications**



Submitted To:

Dr. Shailee Bhatia

Assistant Professor

VSIT, VIPS

Submitted By:

Twinkle Handa

11717702022

BCA 5B

(Batch: 2022-2025)

**Vivekananda Institute of Professional Studies – Technical Campus**

**(Affiliated to Guru Gobind Singh Indraprastha University, Delhi)**

# **ACKNOWLEDGEMENT**

It is our proud privilege to express our profound gratitude to the entire management of Vivekananda Institute of Professional Studies – Technical Campus and teachers of the institute for providing us with the opportunity to avail the excellent facilities and infrastructure. The knowledge and values inculcated have proved to be of immense help at the very start of my career. Special thanks to Hon’ble Founder, Vivekananda Institute of Professional Studies-Technical Campus, Delhi for having provided us an excellent infrastructure at VSIT. I am grateful to Dr. Deepali Kamthania (Dean, VSIT), and Ms. Shailee Bhatia for her astute guidance, constant encouragement and sincere support for this project work. Sincere thanks to all my family members, seniors and friends for their support and assistance throughout the project.

Twinkle Handa 11717702022

# TABLE OF CONTENTS

<b>Acknowledgement .....</b>	<b>1</b>
<b>Table of Contents .....</b>	<b>2</b>
<b>Certificate of the Project Guide .....</b>	<b>3</b>
<b>Self-Certificate .....</b>	<b>4</b>
<b>Synopsis .....</b>	<b>5</b>
<b>Chapter 1: Introduction .....</b>	<b>8</b>
<b>1.1 Objective &amp; Scope of the Project .....</b>	<b>8</b>
<b>1.2 Theoretical Background Definition of Problem .....</b>	<b>9</b>
<b>1.3 System Analysis &amp; Design vis-a-vis User Requirements.....</b>	<b>10</b>
<b>1.4 System Planning (GANTT Chart) .....</b>	<b>11</b>
<b>1.5 Methodology Adopted, System Implementation &amp; Details of Hardware &amp;     Software used.....</b>	<b>12</b>
<b>1.6 System Maintenance &amp; Evaluation .....</b>	<b>14</b>
<b>Chapter 2: Detailed Life Cycle.....</b>	<b>17</b>
<b>2.1 ERD, DFD .....</b>	<b>17</b>
<b>2.2 Input and Output Screen Design.....</b>	<b>20</b>
<b>Chapter 3: Coding and Screenshots.....</b>	<b>22</b>
<b>Chapter 4: Conclusion and Future Scope.....</b>	<b>58</b>
<b>Chapter 5: References.....</b>	<b>60</b>

# **CERTIFICATE**

This is to certify that the project entitled “StockPulse” submitted in partial fulfilment of the requirements for the degree of Bachelor of Computer Applications to Vivekananda Institute of Professional Studies – Technical Campus through VSIT, is an authentic work carried out by Ms. Twinkle Handa, Roll No. 11717702022, under the guidance of Dr. Shailee Bhatia. The matter embodied in this project has not been submitted earlier for the award of any degree, to the best of my knowledge and belief.

Signature of the student

Signature of the Guide

# SELF CERTIFICATE

This is to certify that the project report entitled “StockPulse” done by me is an authentic work carried out for the partial fulfilment of the requirements for the award of the degree of Bachelor of Computer Applications under the guidance of Dr. Shailee Bhatia. The matter embodied in this project work has not been submitted earlier for award of any degree or diploma to the best of my knowledge and belief.

Twinkle Handa

11717702022

# SYNOPSIS

## **Introduction**

The stock market is a complex and dynamic ecosystem where investors buy and sell shares of publicly traded companies. Navigating this intricate landscape requires access to accurate information, powerful analytical tools, and the ability to make informed decisions quickly. This platform will provide a centralized and user-friendly environment where users can access real-time and historical stock data, utilize sophisticated analysis tools, and leverage the predictive power of machine learning models.

## **Problem Statement**

The project addresses the lack of a unified platform for investors to analyze stock data from different perspectives (technical and fundamental analysis). It aims to solve the problem of scattered information and the lack of actionable insights for real-time decision-making in the stock market.

## **Objective and Scope of the Project**

### **Objective:**

This project aims to develop a user-friendly website that will provide a platform for users to access real-time and historical stock data, utilize analysis tools, and obtain predictions generated by ML models.

## Scope:

- **Frontend:** A user-friendly dashboard developed using Streamlit.
- **Backend:** Integration of Alpha Vantage API for stock data, StockNews API for fetching news, and OpenAI for generating insights.
- **Data Visualization:** Use of Plotly for real-time data visualization of stock movements and key financial data.
- **Tools:** Python libraries including pandas, Plotly, Streamlit, and external APIs like Alpha Vantage and OpenAI.
- **Modelling:** Linear regression machine learning model adopted to predict stock prices.

## Methodology

**Agile Methodology:** The project follows an iterative development approach with cycles of development and testing.

1. **Requirement Analysis:** Understanding user needs and components.
2. **System Design:** Creating the backend architecture for data retrieval and processing.
3. **Modelling:** Model training and evaluation using scikit-learn (Linear Regression).
4. **Implementation:** Using APIs to fetch and process stock data, and display it via the Streamlit interface.
5. **Testing:** Ensuring data accuracy, API integration, and UI responsiveness.

## **Hardware & Software to be Used**

- **Hardware:** Basic computing device with internet access.
- **Software:**
  - Programming Language: Python 3.x
  - Operating System: Windows
  - Libraries: pandas, NumPy, Plotly, OpenAI, StockNews, scikit-learn, pandas\_ta, sklearn
  - Frontend: Html, CSS, Streamlit

## **Testing Technologies Used**

- **Unit Testing:** Testing individual components (e.g., fetching stock data, news data, etc.).
- **Integration Testing:** Ensuring APIs work together seamlessly.
- **User Acceptance Testing (UAT):** Testing usability with users to meet requirements.

## **Impact and Conclusion**

The project contributes a valuable tool for investors of all levels to make more informed investment decisions. It provides a user-friendly platform combined with machine learning with integrated data sources, technical analysis, and AI-powered insights, potentially leading to better returns in the stock market.



# CHAPTER 1: INTRODUCTION

StockPulse is a platform that provides a centralized and user-friendly environment where users can access real-time and historical stock data, utilize sophisticated analysis tools, and leverage the predictive power of machine learning models. Chapter 1 introduces StockPulse, outlining its core objectives. It defines the problem of information scarcity and the need for a unified investment research environment. The chapter also covers system analysis and design, aligning user requirements with platform functionalities. A GANTT chart illustrates the project timeline, and the chapter concludes by detailing the Agile methodology, system implementation, and hardware/software tools utilized.

## 1.1. Objective & Scope of the Project

### **Objective:**

The main objective of this project is to build a **Stock Market Analysis** using a combination of financial data, stock news, and technical analysis to help users make informed decisions in the stock market. The project aims to offer users insights into stock pricing trends, fundamental financial data, and news, and provide advanced technical analysis and forecasting capabilities.

Key objectives include:

- **Stock Data Retrieval:** Fetch real-time and historical stock price data from Alpha Vantage.
- **News Aggregation:** Gather and analyze the top 10 latest news related to selected stocks.
- **Technical Analysis:** Visualize stock price movements, technical indicators, and financial metrics.

- **User Insights:** Offer suggestions like reasons to buy or sell stock, and provide SWOT analysis based on data.
- **Prediction:** Use of Machine Learning model to predict stock market prices.

#### Scope:

- **Frontend:** A user-friendly dashboard developed using **Streamlit**.
- **Backend:** Integration of **Alpha Vantage API** for stock data, **StockNews API** for fetching news, and **OpenAI** for generating insights.
- **Data Visualization:** Use of **Plotly** for real-time data visualization of stock movements and key financial data.
- **Tools:** Python libraries including pandas, Plotly, streamlit, and external APIs like **Alpha Vantage** and **OpenAI**.

## **1.2. Theoretical Background & Definition of Problem**

### **Theoretical Background:**

Stock Market Analysis tools use a combination of **technical analysis** and **fundamental analysis** to help investors make informed decisions. The technical analysis uses past price data and patterns to predict future stock movements, while fundamental analysis looks at company financials to assess stock value.

### **Definition of the Problem:**

- **Problem 1:** Investors lack a unified platform for analysing stock data from different perspectives.

- **Problem 2:** Real-time decision-making in the stock market is often hindered due to the scattered nature of information and lack of actionable insights.
- **Problem 3:** Current stock analysis tools are often complex and do not integrate news and social sentiment into their forecasting.

### **1.3. System Analysis & Design vis-a-vis User Requirements**

#### **User Requirements:**

- **User Input:** The user should be able to input a stock ticker (e.g., AAPL for Apple) and a date range for the analysis.
- **Real-Time Data:** The system should retrieve historical stock data and visualize price movements in real-time.
- **Stock Data Visualization:** Display stock trends (like closing price, volume) over the specified date range.
- **News Analysis:** Display the latest news related to the stock with sentiment analysis.
- **Technical Analysis:** Users should have access to various technical indicators (like Moving Averages, RSI, etc.).
- **OpenAI Integration:** Provide reasoning for buying/selling decisions and a SWOT analysis based on the stock data.
- **System Flexibility:** The system should be flexible enough to analyze any stock and its associated data.

**System Design:**

- **Frontend:** Streamlit interface for user input and dynamic content generation.
- **Backend:** API integrations to fetch stock price data, financial data, news, and generate insights via OpenAI.
- **Data Source:** CSV file containing historical stock data (FINAL\_FROM\_DF.csv).
- **Prediction Model:** Linear Regression model trained on the historical data.

**1.4. System Planning (GANTT Chart)**

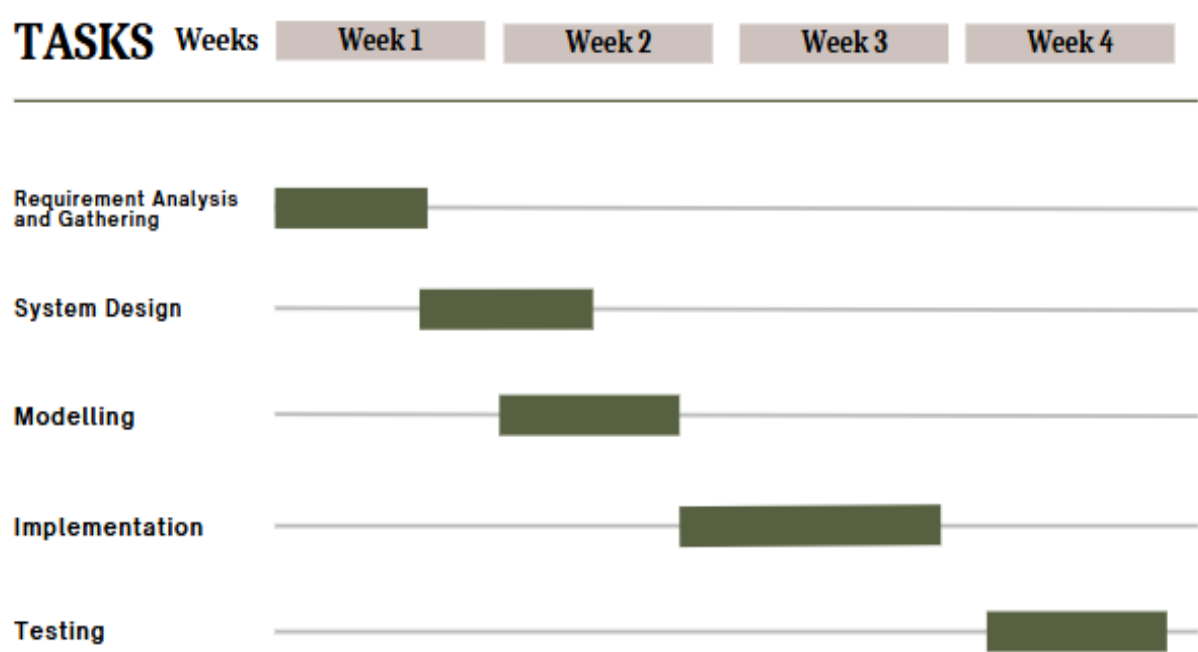


Figure 1: Gantt chart

The chart is organized by weeks, with each task represented by a horizontal bar. The length of the bar indicates the duration of the task.

- **Sequential and Overlapping Tasks:** Some tasks, like System Design and Modelling, happen concurrently. Others, like Implementation, depend on the completion of earlier tasks.
- **Emphasis on Implementation:** The Gantt chart highlights that Implementation is the most time-consuming phase, indicating it's a critical part of the project.
- **Early Testing:** Testing begins relatively early in the process, suggesting an iterative development approach where testing is integrated throughout.

This Gantt chart provides a clear and concise overview of the project schedule, allowing stakeholders to understand the timeline and dependencies between tasks. It's a useful tool for planning, monitoring progress, and ensuring the project stays on track.

## **1.5. Methodology Adopted, System Implementation & Details of Hardware & Software Used**

This section of the report details the practical approaches and tools used to bring the stock market analysis platform to life.

### **Methodology:**

- **Agile Methodology:** This approach was used for the iterative development of the project. Each feature was developed and tested in cycles.
  1. **Requirement Analysis:** Understanding user needs and the necessary components for the dashboard.

2. **System Design:** Creating the backend architecture for data retrieval and processing.
3. **Modelling:** Model training and evaluation using Scikit-learn (Linear Regression).
4. **Implementation:** Using APIs to fetch stock data, process it, and display it via the Streamlit interface.
5. **Testing:** Ensuring data accuracy, proper integration of APIs, and UI responsiveness.

### **System Implementation:**

- **Backend:**
  - **Alpha Vantage API** for fetching stock price data.
  - **StockNews API** for fetching and analysing the latest news.
  - **OpenAI API** for generating stock analysis insights.
  - **Pandas** for data manipulation and processing.
  - **Sklearn** for machine learning model training.
- **Frontend:**
  - **Streamlit** for developing the interactive user interface.
  - **Plotly** for visualizing stock price trends.
  - **Python Libraries:** pandas, NumPy, plotly, openai, stocknews, sklearn, pandas\_ta for technical analysis.

## Hardware & Software:

- **Hardware:** Basic computing device with internet access.
- **Software:**
  - **Programming Language:** Python 3.x Libraries
  - **Operating System:** Windows

## 1.6. System Maintenance & Evaluation

Once the stock market analysis platform is deployed, ongoing maintenance and evaluation are essential to ensure its continued accuracy, reliability, and user satisfaction. Here's an outline of key maintenance and evaluation activities:

### Maintenance

- **Regular Updates:**
  - **Data Sources:** Continuously monitor the APIs (Alpha Vantage, StockNews) for any changes or updates. Adapt the system to accommodate any API revisions to ensure uninterrupted data flow.
  - **Software Libraries:** Keep the Python libraries (pandas, Plotly, scikit-learn, etc.) up-to-date to benefit from bug fixes, performance improvements, and new features.
  - **Streamlit Framework:** Stay current with Streamlit framework updates to maintain compatibility and leverage enhancements for the user interface.

- **Data Monitoring:**
  - **Accuracy:** Regularly check the accuracy of the fetched stock data and news by comparing it with alternative sources. Implement data validation checks to identify and address any discrepancies.
  - **Model Performance:** Continuously evaluate the machine learning model's performance by monitoring its predictive accuracy. Retrain the model periodically with new data to maintain its effectiveness and prevent model drift.
- **Bug Fixes and Troubleshooting:**
  - **Error Logging:** Implement a system to log errors and exceptions encountered during platform operation. Regularly review error logs to identify and resolve bugs or technical issues.
  - **User Feedback:** Provide a mechanism for users to report bugs or technical difficulties. Actively address user-reported issues to ensure a smooth user experience.

## Evaluation

- **Usability Testing:**
  - **User Feedback:** Conduct periodic usability testing sessions with users to gather feedback on the platform's ease of use, intuitiveness, and overall user experience.
  - **A/B Testing:** Experiment with different UI designs or features using A/B testing to determine which variations lead to improved user engagement and satisfaction.



- **Performance Monitoring:**

- **System Responsiveness:** Monitor the platform's performance in terms of loading times, responsiveness to user actions, and overall efficiency. Optimize code and infrastructure to maintain optimal performance.
- **Scalability:** Evaluate the platform's ability to handle increasing amounts of data and user traffic. Plan for scalability to accommodate future growth and prevent performance degradation.

- **Impact Assessment:**

- **User Satisfaction:** Track user satisfaction through surveys, feedback forms, or user engagement metrics. Assess whether the platform is meeting user needs and expectations.
- **Investment Outcomes:** While challenging to measure directly, attempt to gather data on how the platform is influencing user investment decisions and outcomes. This could involve user testimonials, case studies, or tracking the performance of model-generated predictions.

By consistently maintaining and evaluating the stock market analysis platform, you can ensure its long-term viability, accuracy, and user satisfaction. This commitment to ongoing improvement will contribute to the platform's success in empowering investors to make informed decisions in the stock market.

# CHAPTER 2: DETAILED LIFE CYCLE

Chapter 2 details the system's life cycle, including an ER diagram illustrating entities and relationships, a DFD showing data flow, input/output screen designs, data processing steps, and testing methodologies.

## 2.1. ERD (Entity-Relationship Diagram)

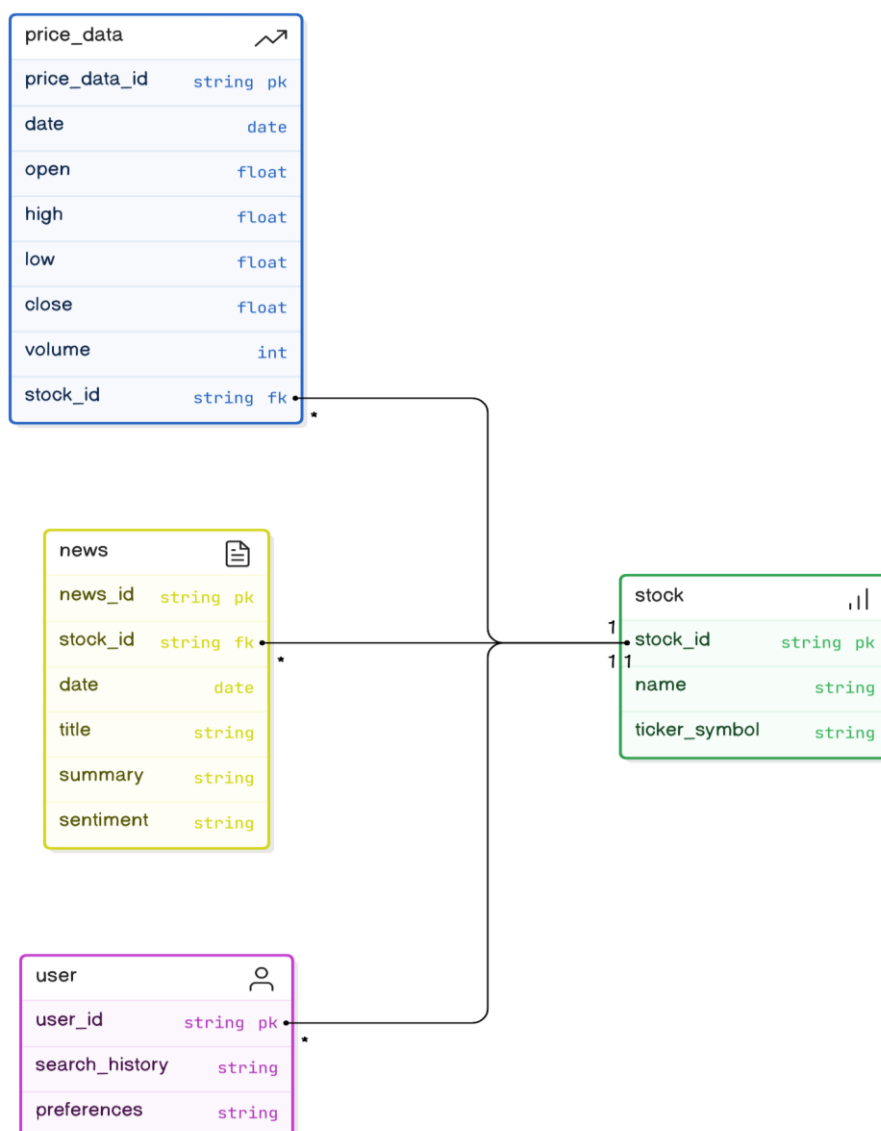


Figure 2: Entity-Relation Diagram

This ER diagram helps in understanding the data structure of the stock market analysis system. It clarifies how different data elements are connected and organized, which is essential for designing and implementing the database and the application logic.

### **Entities:**

- **price\_data:** This entity stores historical stock price information.
  - Attributes include price\_data\_id, date, open, high, low, close, and volume. The stock\_id acts as a foreign key, linking it to the stock entity.
- **news:** This entity holds news articles related to the stock market.
  - Attributes include news\_id, stock\_id, date, title, summary, and sentiment.
- **stock:** This entity represents individual stocks.
  - Attributes include stock\_id, name, and ticker\_symbol.
- **user:** This entity represents users of the system.
  - Attributes include user\_id, search\_history, and preferences.

### **Relationships:**

- The lines connecting the entities represent relationships and how they interact.
- **price\_data and stock:** A price\_data record belongs to a specific stock, indicating a one-to-many relationship.
- **news and stock:** A news article is related to a specific stock, also a one-to-many relationship.

## DFD (Data Flow Diagram)

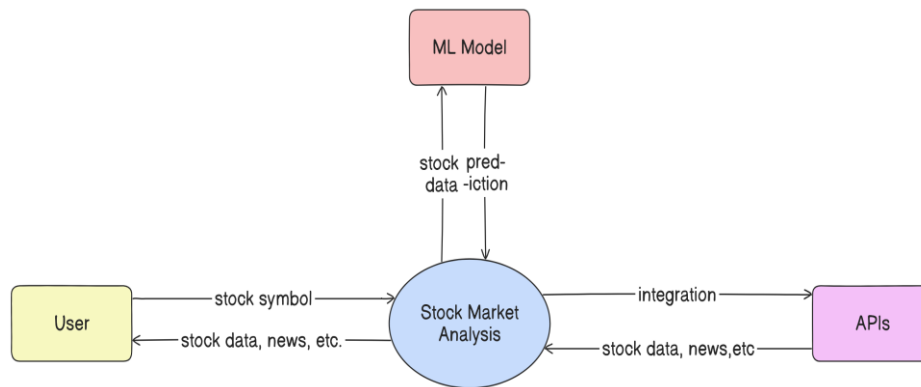


Figure 3: Data Flow Diagram

This Data Flow Diagram, also known as a context diagram, provides a high-level overview of the entire stock market analysis system and its interactions with external entities. Here's a breakdown:

### **Process:**

- **Stock Market Analysis:** This central process represents the core functionality of the system. It encompasses all the activities involved in analysing stock market data, generating predictions, and providing insights to the user.

### **External Entities:**

- **User:** The person interacting with the system to analyse stocks and obtain predictions.
- **APIs:** External Application Programming Interfaces that provide data to the system. This includes APIs like Alpha Vantage (for stock data) and StockNews (for financial news).
- **ML Model:** This represents the machine learning model used for making stock price predictions.

### **Data Flows:**

- **stock symbol:** The user provides the system with a stock symbol (e.g., AAPL, MSFT) as input.
- **stock data, news, etc.:** The APIs provide the system with various data, including historical stock prices, financial news and then the system sends processed data, analysis results, and visualizations back to the user.
- **stock prediction:** The system provides stock price predictions to the ML Model for evaluation or further processing.
- **stock prediction data:** The ML Model sends data or feedback related to stock predictions back to the system.

## **2.2 Input and Output Screen Design**

- **Input Screen:**
  - Home Page
  - Main Menu – Analysis and Prediction
  - Analysis
    - Text input for stock ticker
    - Date range selectors for start and end date
    - Submit button for fetching data
  - Prediction
    - Text input for stock ticker
    - Submit button for fetching data
- **Output Screen:**

- Analysis
  - Line chart for stock prices
  - Data tables for stock prices and fundamental data
  - News headlines with sentiment analysis
  - Buy/Sell reasoning and SWOT analysis
- Prediction
  - Predicted future price
  - Best estimated Linear Regression line

#### **d. Processing Involved**

- **Data Retrieval:** Use of the Alpha Vantage API to fetch stock data and StockNews API for the latest news.
- **Technical Analysis:** Use of pandas\_ta for calculating technical indicators like moving averages, RSI, etc.
- **Sentiment Analysis:** Using the StockNews API and OpenAI API to analyze the sentiment of the stock news and generate insights for users.

#### **e. Methodology Used Testing**

- **Unit Testing:** Testing individual components (e.g., fetching stock data, news data, etc.).
- **Integration Testing:** Ensuring the APIs work together and data flows seamlessly from one part of the system to another.
- **User Acceptance Testing (UAT):** Testing the system's usability with a set of users to make sure the app meets the requirements and is user-friendly.

# CHAPTER 3: CODING AND SCREENSHOTS

This chapter provides the code implementation and illustrative screenshots of StockPulse

## Frontend

### home.html

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title> StockPulse </title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <div class="header-container">

        <h1 class="title">StockPulse<br></h1>

        <h4>Empowering your investments</h4>

        <button><a href='menu.html'>Let's Get Started</a></button>

    </div>

    <footer>

        <p>&copy; 2024 StockPulse </p>

    </footer>

</body>

</html>
```

## **style.css**

```
body{

    font-family: sans-serif;

    background-image:url("stock.jpeg");

    background-size: cover;

}

footer {

    background-color: #333;

    color: white;

    text-align: center;

    padding: 10px;

    position: fixed;

    bottom: 0;

    width: 100%;

}

.title{

    text-align: center;

    position: absolute;

    top: 22%;

    left: 50%;

    color:white;

    transform: translate(-50%, -50%);

    font-size: 3em;

    margin-bottom: 10px;
```



```
}  
  
h4{  
  
    text-align: center;  
  
    position: absolute;  
  
    top: 29%;  
  
    left: 50%;  
  
    color:white;  
  
    transform: translate(-50%, -50%);  
  
    font-size: 1.7em;  
  
    margin-bottom: 10px;  
  
}  
  
button {  
  
    background-color: light grey;  
  
    position: absolute;  
  
    top: 40%;  
  
    left: 42%;  
  
    border: none;  
  
    color: white;  
  
    padding: 15px 32px;  
  
    text-align: center;  
  
    text-decoration: none;  
  
    display: inline-block;  
  
    font-size: 16px;  
  
    margin: 4px 2px;
```

```
    cursor: pointer;

    border-radius: 20px;
}

button: hover {

    background-color: white;

    transform: scale(1.1);
}

button a{

    color: black;

    text-decoration: none;
}
```

### **navfooter.css**

```
.navbar {

    background-color: #333;

    padding: 10px;

    overflow: hidden;

    text-align: center;

    position: fixed;

    top: 0;

    left: 0;

    width: 100%;

    z-index: 100;
}
```

```
.navbar a {  
  
    float: left;  
  
    color: white;  
  
    text-align: center;  
  
    padding: 14px 16px;  
  
    text-decoration: none;  
  
    font-size: 1em;  
  
}  
  
.navbar a:hover {  
  
    background-color: #ddd;  
  
    color: black;  
  
}  
  
footer {  
  
    background-color: #333;  
  
    color: white;  
  
    text-align: center;  
  
    padding: 10px;  
  
    position: fixed;  
  
    bottom: 0;  
  
    width: 100%;  
  
}
```

## menu.html

```
<!DOCTYPE html>

<html>

<head>

  <title>StockPulse</title>

  <link rel="stylesheet" href="menustyle.css">

  <link rel="stylesheet" href="navfooter.css">

</head>

<body>

  <nav class='navbar'>

    <a href='home.html'>Home</a>

    <a href='menu.html'>Menu</a>

    <a href='aboutus.html'>About</a>

    <a href='contact.html'>Contact Us</a>

  </nav>

  <header>

    <h1>How Can I Help You?</h1>

    <h6>Get real-time stock analysis, predictions, and news in one
place.</h6>

  </header>

  <nav class="full-page-menu">

    <ul>

      <li><a href="http://localhost:8501/">Analysis</a></li>

      <li><a href="http://localhost:8080/">Prediction</a></li>
```

```
        </ul>

    </nav>

    <footer>

        <p>&copy; 2023 StockPulse </p>

    </footer>

</body>

</html>
```

### **menustyle.css**

```
body {

    font-family: sans-serif;

    margin: 0;

    background-color: #f4f4f4;

}

header {

    color: rgb(38, 41, 85);

    text-align: center;

    font-size: 1.6em;

    position: relative;

    padding-top: 100px;

    background-color: #e9ecef;

    border-bottom: 1px solid #ddd;

}
```

```
.full-page-menu {  
  
  display: flex;  
  
  flex-direction: column;  
  
  align-items: center;  
  
  justify-content: center;  
  
  text-align: center;  
  
  position: relative;  
  
  height: 200px;  
  
  width: 100%;  
  
  gap: 20px;  
  
}  
  
.full-page-menu a {  
  
  text-decoration: none;  
  
  color: white;  
  
  font-size: 1.5em;  
  
  background-color: #3a3f86;  
  
  padding: 15px 30px;  
  
  border-radius: 20px;  
  
  transition: transform 0.2s ease, box-shadow 0.2s ease;  
  
  box-shadow: 0px 2px 5px rgba(0, 0, 0, 0.2);  
  
}  
  
.full-page-menu a:hover {  
  
  transform: scale(1.05);  
  
  box-shadow: 0px 8px 15px rgba(0, 0, 0, 0.3); }
```

```
.full-page-menu ul {  
    list-style: none;  
    margin: 0;  
    padding: 0;  
}  
  
.full-page-menu li {  
    color: white;  
    padding: 10px 20px;  
    text-align: center;  
    text-decoration: none;  
    display: inline-block;  
    font-size: 1.2em;  
    margin: 10px 0px;  
    cursor: pointer;  
    border-radius: 20px;  
}  
  
li: hover {  
    transform: scale(1.1);  
}
```

### **aboutus.html**

```
<!DOCTYPE html>  
  
<html lang="en">  
  
<head>
```

```

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title> StockPulse - About Us</title>

<link rel="stylesheet" href="aboutusstyle.css">

<link rel="stylesheet" href="navfooter.css">

</head>

<body>

  <nav class='navbar'>

    <a href='a.html'>Home</a>

    <a href='menu.html'>Menu</a>

    <a href='aboutus.html'>About</a>

    <a href='contact.html'>Contact Us</a>

  </nav>

  <main>

    <section class="about-us">

      <h1 class='maintitle' >About Us</h1><br>

      <div class="about-content">

        <div class="text-section">

          <p>We are a team of dedicated developers passionate about
empowering investors with the knowledge and tools to make informed
decisions in the stock market.</p>

          <p>Our mission is to provide accurate and reliable stock market
predictions and analysis, leveraging cutting-edge technology and data-
driven insights.</p>

```



<p>We believe that everyone should have access to the resources they need to navigate the complexities of the stock market and achieve their financial goals.</p>

</div>

<div class="team-section">

<div class="member">



<p>Twinkle Handa</p>

</div>

<div class="member">



<p>Shreshth Kindra</p>

</div>

</div>

</div>

</section>

</main>

<footer>

<p>&copy; 2024 StockPulse </p>

</footer>

</body>

</html>

### aboutusstyle.css

```
body{

    font-family: sans-serif;

    padding-top: 50px;

    background-color: #f4f4f4;

    margin:0;

}

.about-us {

    margin-top: 80px;

    color: rgba(0, 0, 0, 0.7);

    margin-top: 60px;

    padding: 20px;

    border-radius:10px;

    display: flex;

    flex-direction: column;

    justify-content: space-around;

    align-items: center;

    text-align: center;

}

.about-us h1 {

    font-size: 3em;

    margin-bottom: 20px;

    margin-top:10px;

}
```

```
.about-content{

    display: flex;

    gap: 50px;

    justify-content: center;

    flex-direction: row-reverse;

    align-items: flex-start;

}

.text-section {

    width: 50%;

    text-align: left;

    line-height: 1.6;

}

.team-section {

    width: 30%;

    display: flex;

    flex-direction: column;

    align-items: center;

}

.member {

    display: flex;

    flex-direction: column;

    align-items: center;

    margin-bottom: 30px;

}
```

```
.member img {  
  
    width: 70px;  
  
    height: 70px;  
  
    border-radius: 50%;  
  
}
```

### **contact.html**

```
<!DOCTYPE html>  
  
<html lang="en">  
  
<head>  
  
    <meta charset="UTF-8">  
  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  
    <title> StockPulse - Contact Us</title>  
  
    <link rel="stylesheet" href="contactstyle.css">  
  
    <link rel="stylesheet" href="navfooter.css">  
  
</head>  
  
<body>  
  
    <nav class='navbar'>  
  
        <a href='home.html'>Home</a>  
  
        <a href='menu.html'>Menu</a>  
  
        <a href='aboutus.html'>About</a>  
  
        <a href='contact.html'>Contact Us</a>  
  
    </nav>  
  
    <main>
```

```

<section class="contact-us">

  <h1>Contact Us</h1>

  <div class="contact-info">

    <h3>If you have any questions, feedback or inquiries, please feel
free to contact us using the following information:</h3>

    <div class="contact-item">

      <p><a
href="mailto:info@stockmarketpredictor.com">info@stockmarketpredictor.com</
a></p>

    </div>

    <div class="contact-item">

      <p><a href="tel:+91 9876543210">+91-98765-43210</a></p>

    </div>

  </div>

</section>

</main>

<footer>

  <p>&copy; 2024 StockPulse </p>

</footer>

</body>

</html>

```

## contactstyle.css

```
body{

    font-family: sans-serif;

    margin: 0;

    background-color: #f4f4f4;

}

.contact-us {

    padding: 40px;

    border-radius: 10px;

    color: rgba(0, 0, 0, 0.7);

    margin-top: 80px;

    text-align: center;

}

.contact-us h3 {

    margin-bottom: 30px;

}

.contact-info {

    margin-top: 20px;

    display: flex;

    flex-direction: column;

    align-items: center;

}

.contact-info p {

    margin-bottom: 10px;
```

```

}

.contact-info a {

    color: rgba(0, 0, 0, 0.7);

    text-decoration: none;

    font-size:1.1em;

}

.contact-item {

    display: flex;

    align-items: center;

    gap: 20px;

    margin-bottom: 20px;

}

.contact-item img {

    width: 50px;

    height: 50px;

}

.contact-item p {

    font-size: 1.3em;

}

.contact-item a {

    text-decoration: none;

    font-size: 1.2em;}

h1, h3{

    color: rgba(0, 0, 0, 0.7); }

```

```
h1{

    font-size: 3em;

    margin-bottom: 20px;

}
```

## Backend

### predictor.py

```
import pandas as pd

from sklearn.linear_model import LinearRegression

from sklearn.model_selection import train_test_split

from sklearn.metrics import r2_score, mean_squared_error

import matplotlib.pyplot as plt

import streamlit as st

st.set_page_config(layout="wide")

st.markdown(

    """

    <style>

    .main > div {

        padding-top: 0rem !important;

    }

    .main .block-container {

        padding-top: 0rem; }

    """
```



```
body {

    font-family: sans-serif;

    margin: 0;

    padding:0;

}

.stButton>button {

    background-color: #3a3f86;

    color: white;

    border: none;

    padding: 10px 20px;

    margin-top:20px;

    text-align: center;

    text-decoration: none;

    display: inline-block;

    font-size: 16px;

    border-radius: 5px;

}

.stTextInput>div>div>input {

    color: #444;

}

footer {

    position: fixed;

    left: 0;

    bottom: 0;
```

```

        width: 100%;

        background-color: #333;

        color: white;

        text-align: center;

        padding: 10px 0;

    }

</style>

""",

    unsafe_allow_html=True,

)

# Load the data

df = pd.read_csv("C:/Users/hp/mp/FINAL_FROM_DF.csv")

# Preprocess the data

df.drop(columns=['OPEN', 'HIGH', 'LOW', 'LAST', 'PREVCLOSE', 'TOTTRDQTY',
'TOTTRDVAL', 'TOTALTRADES', 'ISIN'], inplace=True)

df.dropna(inplace=True)

df['TIMESTAMP'] = pd.to_datetime(df['TIMESTAMP'])

# Streamlit app

st.title("Stock Predictor")

# Get the stock name from the user

stock_name = st.sidebar.text_input('Ticker')

```

```

if st.button("Let's Predict!"):

    if stock_name in df['SYMBOL'].unique():

        # Filter and preprocess data

        df_stock = df[df['SYMBOL'] == stock_name]

        df_stock = df_stock.sort_values('TIMESTAMP')

        df_stock.reset_index(drop=True, inplace=True)

        df_stock.drop(columns=['SYMBOL'], inplace=True)

        df_stock['PREVIOUS_CLOSE'] = df_stock['CLOSE'].shift(1)

        df_stock.drop(index=df_stock.index[0], axis=0, inplace=True)

        df_stock.drop(columns=['TIMESTAMP'], inplace=True)


        # Split data

        X = df_stock[['PREVIOUS_CLOSE']]

        y = df_stock['CLOSE']

        X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)


        # Train the model

        model = LinearRegression()

        model.fit(X_train, y_train)


        # Make predictions

        y_pred = model.predict(X_test)

```

```

# Evaluate the model

r2 = r2_score(y_test, y_pred)

mse = mean_squared_error(y_test, y_pred)


last_close = df_stock['PREVIOUS_CLOSE'].iloc[-1]

future_price = model.predict([[last_close]])


st.markdown(f"<h3 style='color: #333;'>Predicted future closing price
for {stock_name}: {future_price[0]:.2f}</h3>", unsafe_allow_html=True)


# Plot the best fit line

fig, ax = plt.subplots()

ax.scatter(X_test, y_test, color='blue')

ax.plot(X_test, y_pred, color='red')

ax.set_xlabel('PREVIOUS_CLOSE')

ax.set_ylabel('CLOSE')

ax.set_title('Best Fit Line')

st.pyplot(fig)


else:

    st.write(f"The stock symbol {stock_name} does not exist in the
dataset.")


st.markdown("<footer><p>&copy;      2024      StockPulse      </p></footer>",
unsafe_allow_html=True)

```

## **analysis.py**

```
import streamlit as st

import pandas as pd

from alpha_vantage.timeseries import TimeSeries

from alpha_vantage.fundamentaldata import FundamentalData

import plotly.express as px

from stocknews import StockNews

import openai


st.set_page_config(layout="wide")


st.markdown(

    """

    <style>

    .main > div {

        padding-top: 0rem !important;

    }

    .main .block-container {

        padding-top: 0rem;

    }

    body {

        font-family: sans-serif;

        margin: 0;
```

```

        padding:0;

    }

    footer {

        position: fixed;

        left: 0;

        bottom: 0;

        width: 100%;

        background-color: #333;

        color: white;

        text-align: center;

        padding: 10px 0;

    }

</style>

"",

    unsafe_allow_html=True,

)

# Alpha Vantage API key

api_key = 'MWLJPQIZ9WOWECUW'

# OpenAI API key

openai.api_key = "your-openai-api-key"

```

```

st.title('Stock Analysis')

ticker = st.sidebar.text_input('Ticker', 'AAPL')

start_date = st.sidebar.date_input('Start Date',
pd.to_datetime("2020-01-01"))

end_date = st.sidebar.date_input('End Date', pd.to_datetime('2023-01-
01'))

# Convert start_date and end_date to datetime

start_date = pd.to_datetime(start_date)

end_date = pd.to_datetime(end_date)

# Initialize Alpha Vantage TimeSeries client

if ticker:

    try:

        ts = TimeSeries(key=api_key, output_format='pandas')

        data, meta_data = ts.get_daily(symbol=ticker,
outputsize='full')

        data = data[(data.index >= start_date) & (data.index <=
end_date)]

        if not data.empty:

            fig = px.line(data.reset_index(), x='date', y='4. close',

title=f'{ticker} Closing Prices')

```

```

        st.plotly_chart(fig)

        st.write(data)

    else:

        st.error('No data available for the selected ticker and
date range.')

    except Exception as e:

        st.error(f"Error loading data: {e}")

else:

    st.error('Please enter a valid ticker symbol.')

# Tabs for additional information

pricing_data, fundamental_data, news, openai = st.tabs(

    ["Pricing Data", "Fundamental data", "Top 10 news", "OpenAI
ChatGPT"])

# Pricing Data Tab

with pricing_data:

    st.header("Price Movements")

    st.write(data)

# Fundamental Data Tab

fd = FundamentalData(key=api_key, output_format='pandas')

```



```

with fundamental_data:

    st.subheader('Balance Sheet')

    try:

        balance_sheet, _ = fd.get_balance_sheet_annual(ticker)

        bs = balance_sheet.T[2:]

        bs.columns = list(balance_sheet.T.iloc[0])

        st.write(bs)

    except Exception as e:

        st.error(f"Error loading balance sheet: {e}")


st.subheader('Income Statement')

try:

    income_statement, _ = fd.get_income_statement_annual(ticker)

    is1 = income_statement.T[2:]

    is1.columns = list(income_statement.T.iloc[0])

    st.write(is1)

except Exception as e:

    st.error(f"Error loading income statement: {e}")


st.subheader('Cash Flow Statement')

try:

    cash_flow, _ = fd.get_cash_flow_annual(ticker)

    cf = cash_flow.T[2:]

```

```

        cf.columns = list(cash_flow.T.iloc[0])

        st.write(cf)

    except Exception as e:

        st.error(f"Error loading cash flow statement: {e}")

# News Tab

with news:

    try:

        st.header(f'News of {ticker}')

        sn = StockNews(ticker, save_news=False)

        df_news = sn.read_rss()

        for i in range(10):

            st.subheader(f'News {i+1}')

            st.write(df_news['Published'][i] if 'Published' in
df_news.columns else "No published date available")

            st.write(df_news['Title'][i])

            st.write(df_news['Summary'][i])

            st.write(f"Title      Sentiment:
{df_news.get('Sentiment_title', ['No sentiment'])[i]}")

            st.write(f"News      Sentiment:
{df_news.get('Sentiment_summary', ['No sentiment'])[i]}")

    except Exception as e:

```

```

        st.error(f"Error loading news data: {e}")

# OpenAI ChatGPT Tab

def get_chatgpt_response(prompt):

    try:

        response = openai.ChatCompletion.create(

            model="gpt-3.5-turbo",

            messages=[{"role": "system", "content": "You are a helpful
assistant."},

                        {"role": "user", "content": prompt}]

        )

        return response['choices'][0]['message']['content']

    except Exception as e:

        return f"Error in OpenAI API: {e}"

with openail:

    buy_reason, sell_reason, swot_analysis = st.tabs(['3 reasons to
buy', '3 reasons to sell', 'SWOT Analysis'])

    with buy_reason:

        st.subheader(f'3 Reasons to Buy {ticker} Stock')

        st.write(get_chatgpt_response(f"Give 3 reasons to buy {ticker}
stock"))

    with sell_reason:

```

```

        st.subheader(f'3 Reasons to Sell {ticker} Stock')

        st.write(get_chatgpt_response(f"Give 3 reasons to sell
{ticker} stock"))

    with swot_analysis:

        st.subheader(f'SWOT Analysis of {ticker} Stock')

        st.write(get_chatgpt_response(f"Provide a SWOT analysis of
{ticker} stock"))


st.markdown("<footer><p>&copy;      2024      StockPulse</p></footer>",
unsafe_allow_html=True)

```

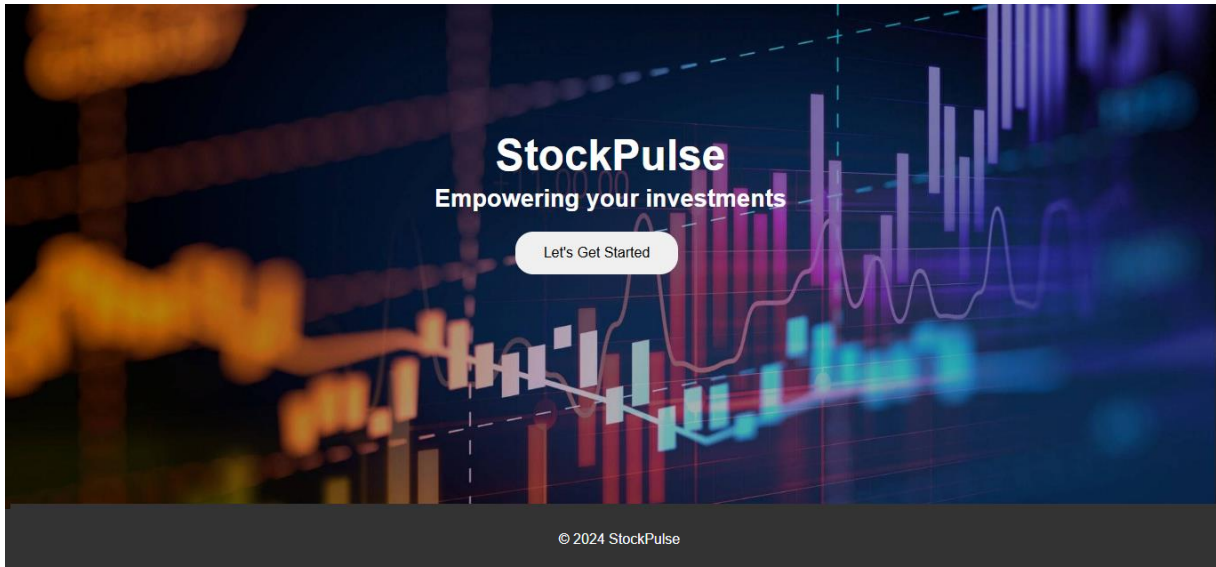


Figure 4: Homepage of 'StockPulse,' a website dedicated to empowering investments

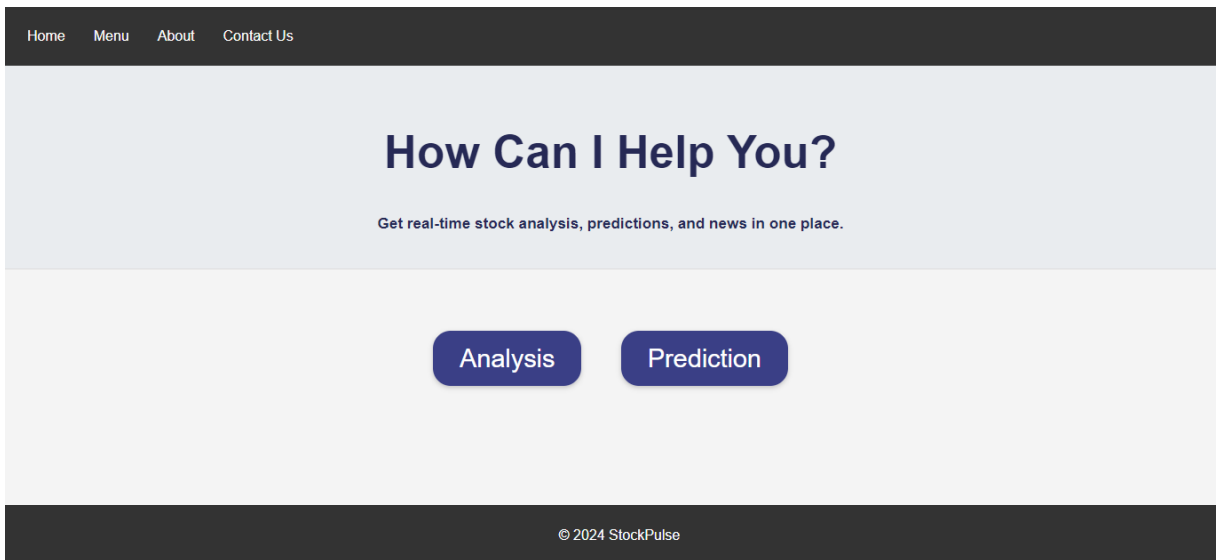


Figure 5: Main menu showcasing the navigation bar to access other pages and the buttons for accessing the site's core functionalities: 'Analysis' and 'Prediction'

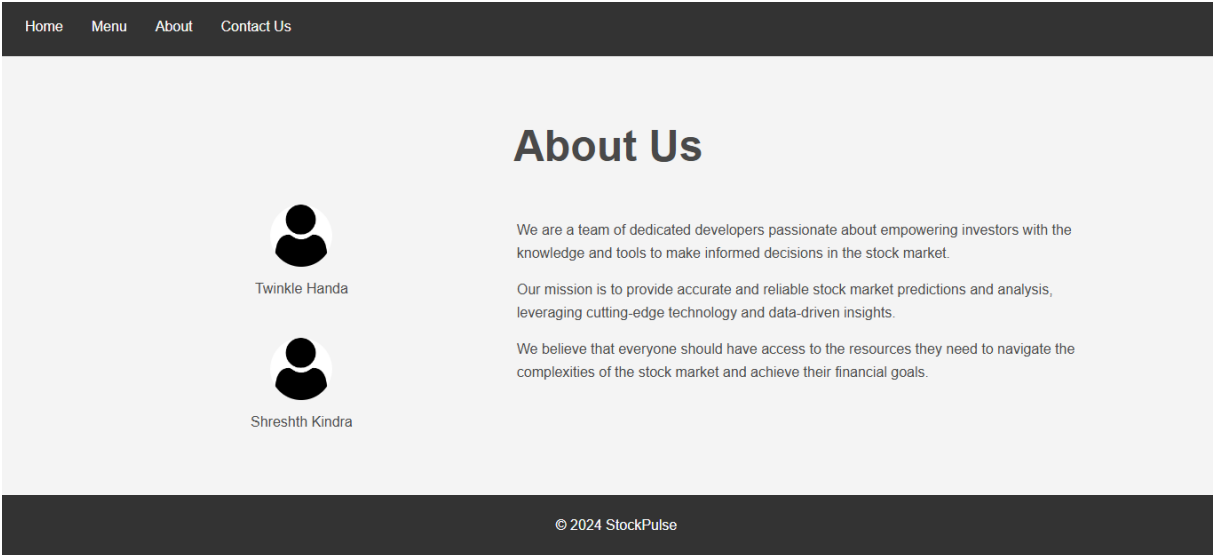


Figure 6: The 'About Us' page introduces the team behind the website, the mission to empower investors, and the belief in accessible resources for navigating the stock market.

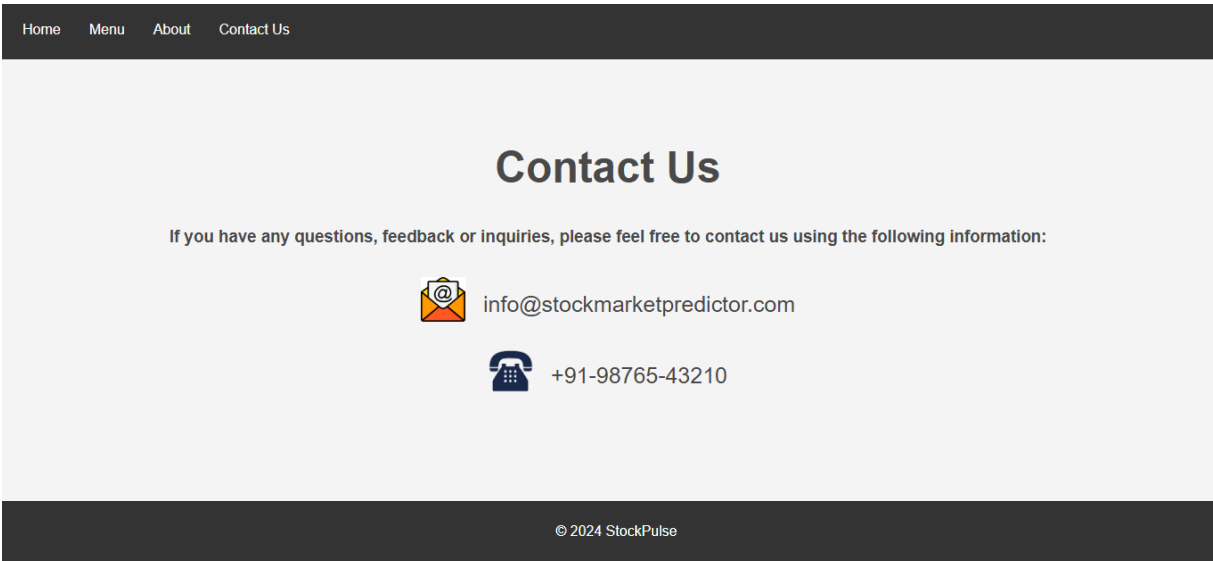


Figure 7: The 'Contact Us' page providing contact information, including email address and phone number, for user inquiries and feedback.

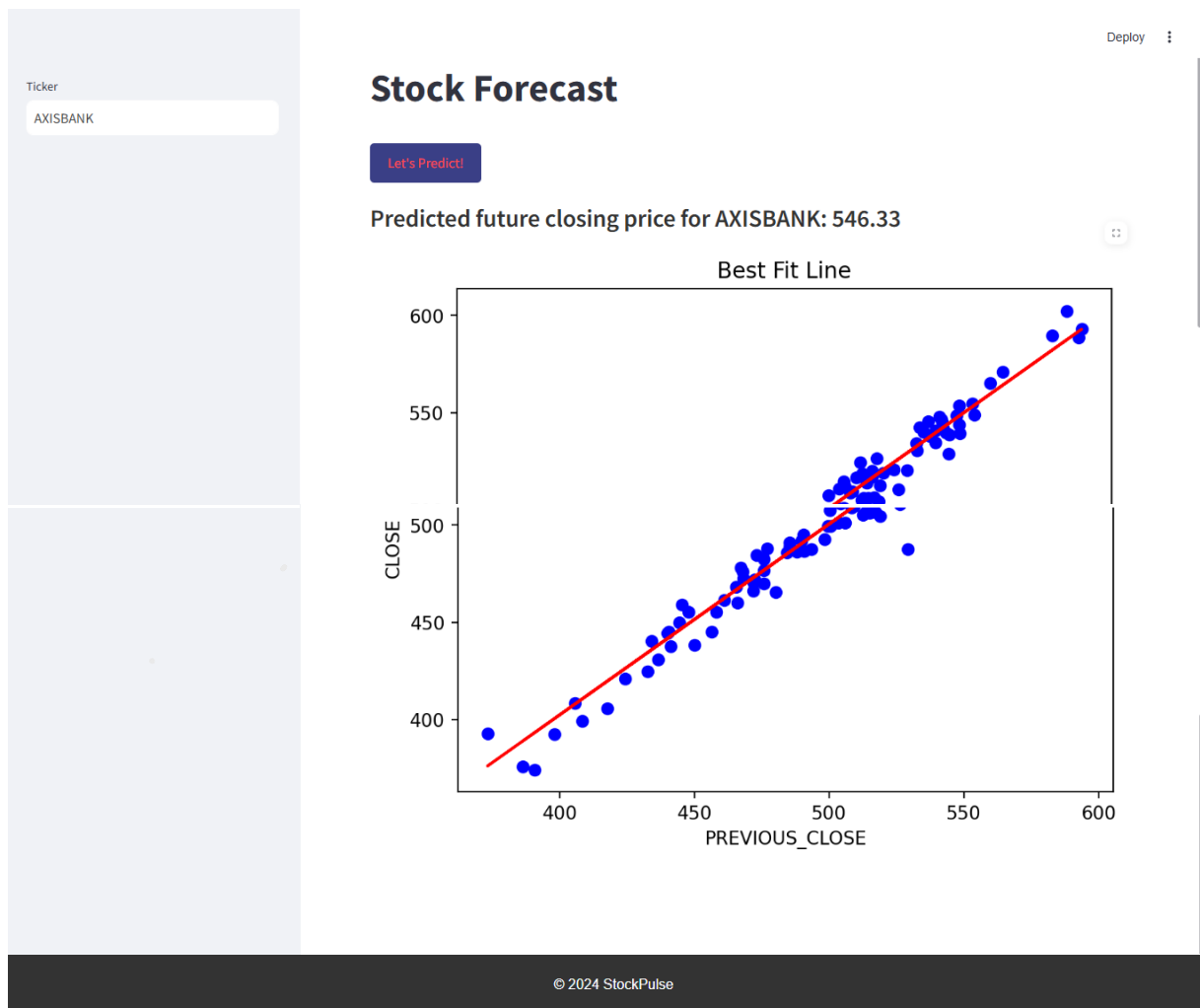


Figure 8: The 'Stock Forecast' page displays the predicted future closing price for shares with a graph illustrating the 'Best Fit Line' based on historical data."





## Price Movements

date	1. open	2. high	3. low	4. close	5. volume
2022-12-30 00:00:00	128.41	129.95	127.43	129.93	77,034,209
2022-12-29 00:00:00	127.99	130.4814	127.73	129.61	75,703,710
2022-12-28 00:00:00	129.67	131.0275	125.87	126.04	85,438,391
2022-12-27 00:00:00	131.38	131.41	128.72	130.03	69,007,830
2022-12-23 00:00:00	130.92	132.415	129.64	131.86	63,814,893
2022-12-22 00:00:00	134.352	134.56	130.3	132.23	77,852,108
2022-12-21 00:00:00	132.98	136.81	132.75	135.45	85,927,993
2022-12-20 00:00:00	131.39	133.25	129.89	132.3	77,432,819
2022-12-19 00:00:00	135.11	135.2	131.32	132.37	79,592,614
2022-12-16 00:00:00	136.685	137.65	133.73	134.51	160,156,900

Figure 10: The 'Pricing Data' tab of the 'Stock Analyzer' page presents a table of historical price movements for a particular stock

## Balance Sheet

	2024-09-30	2023-09-30	2022-09-30	2021-09-30	2020-09-30
intangibleAssetsExcludingGoodwill	None	None	None	None	None
goodwill	None	None	None	None	None
investments	254763000000	264965000000	292870000000	318994000000	345000000000
longTermInvestments	91479000000	100544000000	120805000000	127877000000	100000000000
shortTermInvestments	35228000000	31590000000	24658000000	27699000000	52900000000
otherCurrentAssets	14287000000	14695000000	21223000000	14111000000	11200000000
otherNonCurrentAssets	74834000000	64758000000	54428000000	48849000000	42500000000
totalLiabilities	308030000000	290437000000	302083000000	287912000000	258000000000
totalCurrentLiabilities	176392000000	145308000000	153982000000	125481000000	105000000000
currentAccountsPayable	68960000000	62611000000	64115000000	54763000000	42200000000

## Income Statement

	2024-09-30	2023-09-30	2022-09-30	2021-09-30	2020-09-30
grossProfit	180683000000	169148000000	170782000000	152836000000	104956000000
totalRevenue	391035000000	379352000000	391397000000	363172000000	271642000000
costOfRevenue	236449000000	239069000000	248640000000	234954000000	189475000000
costofGoodsAndServicesSold	210352000000	214137000000	223546000000	212981000000	169559000000
operatingIncome	123216000000	114301000000	119437000000	108949000000	66288000000
sellingGeneralAndAdministrative	26097000000	24932000000	25094000000	21973000000	19916000000
researchAndDevelopment	31370000000	29915000000	26251000000	21914000000	18752000000
operatingExpenses	57467000000	54847000000	51345000000	43887000000	38668000000
investmentIncomeNet	None	3750000000	2825000000	2843000000	3763000000
netInterestIncome	None	-3933000000	-2931000000	-2645000000	-2873000000

## Cash Flow Statement

	2024-09-30	2023-09-30	2022-09-30	2021-09-30	2020-09-30
operatingCashflow	118254000000	110543000000	122151000000	104038000000	80674000000
paymentsForOperatingActivities	1900000000	5703000000	4665000000	4087000000	4502000000
proceedsFromOperatingActivities	None	None	None	None	None
changeInOperatingLiabilities	21572000000	1142000000	15558000000	19801000000	6935000000
changeInOperatingAssets	17921000000	7719000000	14358000000	24712000000	12450000000
depreciationDepletionAndAmortization	11445000000	11519000000	11104000000	11284000000	11056000000
capitalExpenditures	9447000000	10959000000	10708000000	11085000000	7309000000
changeInReceivables	5144000000	417000000	9343000000	14028000000	-8470000000
changeInInventory	1046000000	1618000000	-1484000000	2642000000	1270000000
profitLoss	93736000000	96995000000	99803000000	94680000000	57411000000

Figure 11: The 'Fundamental Data' tab of the 'Stock Analyzer' page displays financial statements for a specific company. It includes detailed tables for the Balance Sheet, Income Statement, and Cash Flow Statement.

# CHAPTER 4: CONCLUSION AND FUTURE

## SCOPE

This chapter summarizes the key findings of the study and discusses potential avenues for future research and development.

### **Conclusion**

The project successfully developed a Stock Market Analysis Dashboard that integrates financial data, stock news, and technical analysis to assist users in making informed investment decisions. By utilizing real-time and historical data, advanced ML models, and a user-friendly web interface, the project achieved its objectives of providing a comprehensive platform for stock market analysis. The dashboard empowers investors of all levels to gain valuable insights into stock pricing trends, fundamental financial data, and relevant news, enabling them to make more informed investment choices.

### **Future Scope**

While the project successfully delivered a functional Stock Market Analysis Dashboard, there are several areas for potential enhancements and future development:

- **Expanded Data Sources:** Integrate additional data sources beyond Alpha Vantage and StockNews API to provide a broader range of information and perspectives. This can include incorporating social media sentiment analysis, economic data, and alternative financial news outlets.
- **Advanced Machine Learning Models:** Explore and implement more sophisticated machine learning models beyond Linear Regression to potentially improve the accuracy

and sophistication of stock price predictions. Consider incorporating deep learning techniques or ensemble methods to capture complex patterns and relationships within the data.

- **Personalized User Experience:** Enhance the user interface and dashboard features to provide a more personalized experience. This can include implementing user profiles, customized watchlists, and the ability to tailor the dashboard to specific investment goals and risk tolerance.
- **Real-time Trading Integration:** Explore the possibility of integrating the dashboard with real-time trading platforms. This would enable users to execute trades directly from the dashboard based on the insights and analysis generated by the tool, streamlining the investment process.
- **Educational Resources:** Incorporate educational resources and interactive tutorials to enhance users understanding of stock market concepts, technical analysis indicators, and investment strategies. This can empower users to make more informed decisions and improve their overall investment literacy.

By addressing these areas for future development, the Stock Market Analysis Dashboard can evolve into an even more comprehensive and valuable tool for investors seeking to navigate the complexities of the stock market and achieve their financial goals.

## CHAPTER 5: REFERENCES

This chapter lists all the sources we used to research and write this report. It includes books, articles, and websites. It serves as a record of the diverse materials consulted, ranging from peer-reviewed journal articles to reputable websites and tools. This chapter acknowledges the intellectual contributions of others and allows readers to trace the origins of the information and ideas discussed throughout the report.

- “Review of Predicting Stock Prices Based on Machine Learning and Stock Learning” by Yi Chao: <https://www.semanticscholar.org/paper/Review-of-Predicting-Stock-Prices-Based-on-Machine-Cao/eb92f157d6651dc4797a647d198e0b8530a39457>
- “Importance of Machine Learning in Making Investment” by Akhilesh Prasad <https://www.semanticscholar.org/paper/Importance-of-Machine-Learning-in-Making-Investment-Prasad-Seetharaman/72c14ef25034f9236d1c075fb02a911c615fe4c7>
- “Real Time Stock Market Analysis” by Naman Adlakha, Ridhima and Avita Katal: <https://ieeexplore.ieee.org/document/9526506>
- “How to make predictions with Linear Regression” by Zach Bobbitt: <https://www.statology.org/predictions-regression/>
- <https://gemini.google.com/gem/learning-coach>
- <https://github.com>
- Dataset Link: <https://www.kaggle.com/datasets/minatverma/nse-stocks-data>
- API: <https://www.alphavantage.co/>  
<https://openai.com/index/openai-api/>  
<https://stocknewsapi.com/>