

# A GAMIFIED AI POWERED INVESTMENT RECOMMENDATION SYSTEM FOR NOVICE INVESTORS



## A DESIGN PROJECT REPORT

*submitted by*

**SARASWATHY P L**

**SUBHIKA S**

**UVASHINI NEKA B S**

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

*in*

**COMPUTER SCIENCE AND ENGINEERING**

**K RAMAKRISHNAN COLLEGE OF TECHNOLOGY**

(An Autonomous Institution, affiliated to Anna University Chennai, Approved by AICTE, New Delhi)

Samayapuram – 621 112

**JUNE 2025**



# A GAMIFIED AI POWERED INVESTMENT RECOMMENDATION SYSTEM FOR NOVICE INVESTORS



## A DESIGN PROJECT REPORT

*submitted by*

**SARASWATHY P L(811722104135)**

**SUBHIKA S(811722104158)**

**UVASHINI NEKA B S(811722104172)**

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

*in*

**COMPUTER SCIENCE AND ENGINEERING**

**K RAMAKRISHNAN COLLEGE OF TECHNOLOGY**

(An Autonomous Institution, affiliated to Anna University Chennai, Approved by AICTE, New Delhi)

Samayapuram – 621 112

**JUNE 2025**

# **K RAMAKRISHNAN COLLEGE OF TECHNOLOGY**

**(AUTONOMOUS)**

**SAMAYAPURAM – 621 112**

## **BONAFIDE CERTIFICATE**

Certified that this project report titled “**A GAMIFIED AI-POWERED INVESTMENT RECOMMENDATION SYSTEM FOR NOVICE INVESTORS**” is Bonafide work of **SARASWATHY P L (811722104135), SUBHIKA S(811722104158), UVASHINI NEKA B S(811722104172)**, who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

### **SIGNATURE**

Dr. A Delphin Carolina Rani, M.E.,Ph.D.,

### **HEAD OF THE DEPARTMENT**

PROFESSOR

Department of CSE

K Ramakrishnan College of Technology

(Autonomous)

Samayapuram – 621 112

### **SIGNATURE**

Mr. R Vignesh Kumar, M.E.,

### **SUPERVISOR**

Assistant Professor

Department of CSE

K Ramakrishnan College of Technology

(Autonomous)

Samayapuram – 621 112

Submitted for the viva-voice examination held on .....

### **INTERNAL EXAMINER**

### **EXTERNAL EXAMINER**

## **DECLARATION**

We jointly declare that the project report on “**A GAMIFIED AI-POWERED INVESTMENT RECOMMENDATION SYSTEM FOR NOVICE INVESTORS**” is the result of original work done by us and best of our knowledge, similar work has not been submitted to “**ANNA UNIVERSITY CHENNAI**” for the requirement of Degree of Bachelor Of Engineering. This project report is submitted on the partial fulfilment of the requirement of the award of Degree of Bachelor Of Engineering.

**Signature**

SARASWATHY P L

---

SUBHIKA S

---

UVASHINI NEKA B S

Place: Samayapuram

Date:

## **ACKNOWLEDGEMENT**

It is with great pride that we express our gratitude and indebtedness to our institution “**K RAMAKRISHNAN COLLEGE OF TECHNOLOGY**”, for providing us with the opportunity to do this project.

We are glad to credit and praise our honorable and respected chairman sir **Dr. K RAMAKRISHNAN, B.E.**, for having provided for the facilities during the course of our study in college.

We would like to express our sincere thanks to our beloved Executive Director **Dr. S KUPPUSAMY, MBA, Ph.D.**, for forwarding our project and offering adequate duration to complete it.

We would like to thank **Dr. N VASUDEVAN, M.Tech., Ph.D.**, Principal, who gave opportunity to frame the project with full satisfaction.

We heartily thank **Dr. A DELPHIN CAROLINA RANI, M.E., Ph.D.**, Head of the Department, **COMPUTER SCIENCE AND ENGINEERING** for providing her support to pursue this project.

We express our deep and sincere gratitude and thanks to our project guide **Mr. R VIGNESH KUMAR , M.E.**, Department of **COMPUTER SCIENCE AND ENGINEERING**, for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

We render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course. We wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

## **ABSTRACT**

This Python application offers a beginner-friendly platform for stock market analysis and financial literacy, integrating real-time data from Alpha Vantage. It serves dual purposes: educating new investors on essential market concepts and enabling practical stock evaluation based on trend analysis, RSI, and predictive profit estimates. Users are guided through foundational knowledge including order types, stock metrics, and risk categories, which help them make informed decisions tailored to their risk tolerance and investment goals.

A gamification layer enhances user engagement through a point and badge system. Actions such as reading guides, watching educational videos, and conducting stock analyses reward users, encouraging continued learning. The profile system tracks accumulated points, unlocked badges, and investment activity, personalizing the experience and creating a sense of progress for novice investors.

The analytical engine evaluates multiple stocks using 14-day price trends and calculates daily price change averages to determine bullish, bearish, or sideways market behavior. These insights inform tailored investment suggestions, calculating optimal amounts to invest based on the user's risk tolerance. Visualizations of stock trends and tabulated reports with RSI and alternative stock recommendations enhance interpretability.

## TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	<b>ABSTRACT</b>	v
	<b>LIST OF FIGURES</b>	ix
	<b>LIST OF ABBREVIATIONS</b>	x
<b>1</b>	<b>INTRODUCTION</b>	
	1.1 Background	1
	1.2 Overview	2
	1.3 Problem Statement	3
	1.4 Objective	3
	1.5 Implication	4
<b>2</b>	<b>LITERATURE SURVEY</b>	6
<b>3</b>	<b>SYSTEM ANALYSIS</b>	12
	3.1 Existing System	12
	3.1.1 Basic Financial Platforms	12
	3.1.2 Professional Tools	12
	3.1.3 Charting And Technical Platform	13
	3.1.4 Mobile Investment Apps	13

3.1.5 Stock Screeneers	13
3.1.6 AI and Algorithmic	14
3.1.7 Disadvantages	14
3.2 Proposed System	14
3.3 Block Diagram	16
3.4 Flow Chart	17
<b>4 MODULES</b>	18
4.1 Module Description	18
4.1.1 Data Processing Module	18
4.1.2 Feature Selection Module	19
4.1.3 Model Selection and Training	20
4.1.4 Performance Metrics Module	23
<b>5 SOFTWARE DESCRIPTION</b>	25
5.1 Software Requirements	25
5.1.1 Operating System	25
5.1.2 Python Environment	26
5.1.3 Python Libraries	26
5.1.4 Built-in Python Libraries	26
5.1.5 Alpha Vantage API Key	27

5.1.6	Code Editors	27
<b>6</b>	<b>TEST RESULT AND ANALYSIS</b>	<b>28</b>
6.1	Testing	28
6.2	Test Objectives	28
<b>7</b>	<b>RESULT AND DISCUSSION</b>	
7.1	Result	30
7.2	Conclusion	31
7.3	Future Enhancement	32
	<b>APPENDIX A (SOURCE CODE)</b>	<b>34</b>
	<b>APPENDIX B (SCREENSHOTS)</b>	<b>43</b>
	<b>REFERENCES</b>	<b>45</b>

## **LIST OF FIGURES**

<b>FIGURE NO</b>	<b>FIGURE NAME</b>	<b>PAGE NO</b>
3.3	Proposed Diagram	16
3.4	Flow chart	17

## LIST OF ABBREVIATIONS

<b>ABBREVIATION</b>	<b>FULL FORM</b>
API	Application Programming Interface
RSI	Relative Strength Index
SMA	Simple Moving Average
EMA	Exponential Moving Average
MACD	Moving Average Convergence Divergence
ETF	Exchange-Traded Fund
IDE	Integrated Development Environment
P/E Ratio	Price-to-Earnings Ratio
EPS	Earnings Per Share
VaR	Value at Risk
AI	Artificial Intelligence
ML	Machine Learning
GUI	Graphical User Interface
LSTM	Long Short-Term Memory
JSON	JavaScript Object Notation
NLP	Natural Language Processing

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 BACKGROUND**

Investing in the stock market has become increasingly accessible due to advancements in financial technology and the availability of online trading platforms. Despite this, many beginners remain hesitant to invest, largely due to a lack of financial literacy, fear of risk, and the overwhelming complexity of market data. Traditional financial tools often cater to experienced investors, using jargon-heavy analytics and providing little to no educational support for newcomers. This creates a significant gap for entry-level users who seek guidance, clarity, and confidence in their investment decisions. In response to this challenge, educational initiatives and fintech applications have begun to incorporate gamified learning and simplified analytics to make investing more approachable. However, many such tools lack interactivity, personalized insights, or direct integration with real-time stock data. Moreover, few combine technical analysis with engaging user experiences or track educational progress, which can be crucial for building lasting investment habits. This project addresses these gaps by developing an all-in-one Python-based solution that blends real-time stock analysis, investor education, and gamification. Using the Alpha Vantage API for accurate market data, the tool allows users to explore price trends, assess risk, and make guided investment decisions based on their personal risk profile and financial balance. Simultaneously, users can access educational resources and receive rewards for learning and analysis, enhancing motivation and retention. By integrating learning, analytics, and engagement, the project aims to transform the investment journey from a daunting process into an empowering and educational experience. It encourages financial literacy through active participation, offering beginner investors a supportive environment to grow their skills and confidence in navigating the stock market.

## 1.2 OVERVIEW

This Python-based stock analysis tool is an interactive console application that empowers beginners to make informed investment decisions through analysis, education, and gamification. Leveraging the Alpha Vantage API, it fetches 14 days of stock data and performs technical analysis including trend detection (Bullish, Bearish, or Sideways), prediction of percentage change, and Relative Strength Index (RSI) calculations. It also generates personalized investment suggestions based on user-defined risk tolerance and balance, while recommending alternative stocks aligned with the current market trend. Stock performance is visualized through plotted trends to help users better understand price movements.

The system also serves as an educational platform for novice investors. It offers a beginner-friendly investment guide and curated video resources covering topics such as stock market basics, chart reading, and technical analysis. Users can choose which videos to watch, earning points for each educational activity. These features aim to build foundational knowledge, encouraging long-term learning and smarter investment choices. The user profile tracks progress, points, and awarded badges—motivating users through achievements like “First Step” and “Market Wizard.”

A key feature of this tool is its gamified user profile system, which records points for various actions such as viewing guides, watching videos, and conducting stock analyses. Badges are awarded when point milestones are reached, and investment history is stored with details like stock symbols, investment amount, and status. This encourages ongoing engagement and creates a rewarding experience for users as they learn and invest. The app's integration of education, visualization, and gamification makes it a powerful tool for anyone starting their investment journey.

## **1.3 PROBLEM STATEMENT**

Many beginner investors struggle to navigate the stock market due to a lack of financial knowledge, limited access to user-friendly analytical tools, and the absence of engaging educational support, which often leads to poor investment decisions or complete avoidance of investing. Existing platforms are typically tailored for experienced users, offering complex data visualizations and terminology that intimidate newcomers. Additionally, there is minimal integration of motivation-driven features such as progress tracking or gamified learning, which are proven to improve user engagement and retention. This project aims to address these challenges by providing a simplified, interactive, and gamified platform that combines real-time stock analysis, personalized investment suggestions, and accessible educational resources. By guiding users through data-driven decision-making and rewarding their learning progress, the tool seeks to make investing approachable, informative, and empowering for first-time and casual investors.

## **1.4 OBJECTIVE**

The primary objective of this project is to develop an interactive, beginner-friendly stock analysis tool that simplifies the complexities of financial markets for novice investors. By leveraging real-time data from the Alpha Vantage API, the tool enables users to analyze multiple stock trends, assess market sentiment, and make informed investment decisions based on personalized risk tolerance and capital. It also calculates predictive insights using price changes and RSI indicators, guiding users with visual aids like trend charts to enhance understanding.

A secondary objective is to integrate educational content that fosters continuous learning. The tool offers a structured investment guide and curated beginner-friendly video tutorials on topics such as stock fundamentals, technical

analysis, and investment strategies. This embedded learning framework ensures that users not only receive actionable insights but also build the financial literacy necessary to sustain long-term investing habits.

Another key goal is to incentivize user engagement through gamification. Users accumulate points for completing key actions such as running analyses, watching videos, and reading the guide. As points accumulate, users unlock badges that serve as milestones, reinforcing a sense of achievement and motivating consistent use of the platform. This feature transforms a traditionally data-heavy experience into an engaging and rewarding journey for users.

Finally, the project aims to personalize the investment experience through features like risk-adjusted investment suggestions and alternative stock recommendations. Each analysis dynamically adapts to the user's risk preference and balance, ensuring tailored advice that resonates with individual financial goals. By combining education, analytics, visualization, and gamification, this tool aspires to democratize investment knowledge and promote financial empowerment among new investors.

## 1.5 IMPLICATION

The development of this interactive stock analysis and educational platform has several significant implications for financial literacy, user engagement, and technology-assisted investing. Firstly, it democratizes access to stock market insights by transforming complex financial data into simplified, visually intuitive information that can be understood by users with little to no prior experience. This can help reduce the entry barriers for novice investors, encouraging wider participation in financial markets and promoting more informed decision-making.

Secondly, the integration of gamification—through points, badges, and educational milestones—adds a motivational layer that can significantly enhance

user retention and learning outcomes. As users engage more deeply with the platform, they build not only their knowledge but also confidence in managing their investments, which is a critical step toward long-term financial independence. This approach also has broader educational implications, suggesting that gamified learning could be effectively used in other areas of personal finance education.

Furthermore, by offering personalized investment suggestions based on market trends and user-specific risk profiles, the tool supports a more responsible and strategic investment approach. This feature, combined with alternative stock recommendations and RSI-based technical indicators, positions the platform as both a learning aid and a decision-support system.

Ultimately, this project serves as a model for how fintech solutions can combine real-time data analytics, user-centric design, and educational value to bridge the gap between interest in investing and the knowledge needed to do it effectively. Its success could inspire further innovation in developing accessible financial tools for underserved populations.

## CHAPTER 2

### LITERATURE SURVEY

#### **1. "Stock Market Prediction Using Machine Learning Algorithms" – J. Patel, S. Shah, P. Thakkar, and K. Kotecha (2015)**

This research provides an in-depth comparison between various machine learning models such as Support Vector Machines (SVM), Artificial Neural Networks (ANN), and Random Forest in the context of predicting stock price movements. The authors utilized historical stock data along with popular technical indicators like RSI, EMA, and MACD. They discovered that Random Forest outperformed the other models in both accuracy and reliability. This paper is significant to our project as it reinforces the relevance of incorporating technical indicators in prediction models. It also confirms the utility of using compact historical data efficiently. In our assistant, the incorporation of RSI and trend detection logic follows a similar philosophy of simplifying complex algorithms for accessibility. Even though our model is rule-based rather than ML-based, it draws from the predictive principles tested here. Moreover, this study supports the rationale for technical indicator use in real-time investment guidance systems. It forms a theoretical backbone for how predictive accuracy can be retained with thoughtful indicator selection.

#### **2. "Predicting the Direction of Stock Market Prices Using Tree-Based Classifiers" – Y. Kara, M. A. Boyacioglu, and Ö. K. Baykan (2011)**

This paper explores the use of Decision Trees and Random Forest classifiers for predicting the up or down movement of stock prices. Using historical pricing and volume data, it concludes that tree-based models offer intuitive and accurate predictions. The significance lies in their interpretability, which is a key

consideration for any financial application. Our project uses similar historical and trend-based data inputs to determine market sentiment (Bullish, Bearish, or Sideways), thus reflecting the principles of this research. While our approach does not involve tree-based classifiers, it adopts the same structured logic to derive decisions. The study is crucial because it shows how directional movement can be predicted with minimal features, supporting our use of SMA and percent change over a short window. The findings underscore the practicality of interpretable systems for real-time decision-making in financial advisory tools, especially those targeted at beginner-level investors.

### **3. "Financial Sentiment Analysis for Risk Prediction on Investment Decisions" – B. Weng, J. Ahmed, and A. Megahed (2017)**

This work addresses how investor behavior and sentiment derived from online data sources can impact risk assessment and investment strategies. The authors argue for a behavioral lens in financial modeling, suggesting that investors' emotions and information literacy strongly influence market dynamics. Our project integrates this principle through a gamified user profile system that encourages learning and engagement. Users gain points, ranks, and badges as they interact with tutorials and content, indirectly modeling their sentiment and awareness. This behavioral metric helps tailor investment suggestions, making the assistant both responsive and educational. Although not using real-time sentiment from external sources, the project replicates the behavioral awareness internally, ensuring that users develop sound strategies. By linking user progress to content delivery and market suggestions, the assistant adopts a practical application of behavioral finance theories discussed in this paper.

### **4. "An Empirical Study on Modeling and Prediction of Bitcoin Prices with Bayesian Neural Networks Based on Blockchain Information" – H. Jang and J. Lee (2018)**

The study presents a forecasting method for Bitcoin prices using Bayesian neural networks and features extracted from blockchain data. The core idea is that meaningful insights can be derived from limited but relevant and recent datasets. This principle is mirrored in our use of Alpha Vantage's compact time-series data (14-day daily) to determine stock market trends. By focusing on high-impact, short-term indicators like RSI and percent change, the assistant ensures timely and practical advice. This study's emphasis on data efficiency and prediction performance is directly applicable to lightweight systems intended for real-time operation. It validates our decision to prioritize accessibility and response time over complex long-term modeling. This shows that intelligent advisory systems can be both minimal and effective when built on strategically chosen indicators.

## **5. "Genetic Algorithms Approach to Feature Discretization in Neural Networks for Prediction of Stock Price Index" – K.-J. Kim and I. Han (2000)**

This pioneering paper explores how genetic algorithms can improve neural network performance by selecting and discretizing optimal input features. The results suggest that well-chosen inputs lead to significant accuracy improvements. Our project, while not involving neural networks or genetic algorithms, similarly emphasizes the careful selection of input features. The use of key technical indicators like change rate, SMA, and RSI reflects a focus on high-impact, low-complexity data. This aligns with the paper's message that model effectiveness is deeply tied to input quality. The assistant benefits from this insight by avoiding noisy or redundant features, instead depending on compact and easily interpretable data points that enhance user trust and comprehension. This enhances system efficiency and keeps predictions clear and explainable.

## **6. "Sentiment Analysis and Its Applications in Stock Market Prediction: A Survey" – H. Chen, P. De, Y. Hu, and B.-H. Hwang (2014)**

This paper comprehensively reviews how sentiment data—extracted from news articles, social media, and blogs—can be incorporated into stock market prediction models. It advocates combining technical and sentiment-based features for improved accuracy. Our current assistant is technically focused but can evolve to include sentiment features in future phases. The assistant's educational content delivery system can serve as an indirect sentiment proxy, reflecting user confidence and knowledge levels. This paper's suggestion to blend different data types inspires a hybrid approach, where behavioral and technical signals converge. It informs our roadmap toward building a more responsive, user-centric investment assistant. The emphasis on alternative data use shows potential future directions for improving investment guidance systems.

## **7. "Application of Reinforcement Learning in Stock Market Trading: A Review" – Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai (2016)**

This paper surveys the application of reinforcement learning (RL) models to stock trading problems, where agents learn optimal trading strategies from experience. While RL systems are powerful, they are complex and require significant training data. Our assistant takes inspiration from the principle of learning from user interactions, but implements it in a lightweight, gamified format. By tracking user engagement and upgrading their profile level, the system simulates a learning loop that rewards investment in learning. The concept of adaptive guidance, drawn from RL, informs our content suggestion engine. Thus, while technically distinct, our model philosophically aligns with RL principles in a simplified and accessible format suitable for non-expert users.

## **8. "A Survey of Financial Applications of Machine Learning" – A. Gu, B. Kelly, and D. Xiu (2020)**

This recent paper presents a broad overview of machine learning applications across financial services, including asset pricing, fraud detection, and portfolio management. It emphasizes the need for explainable and user-aligned AI systems. Our assistant aligns with this emerging paradigm by providing transparent investment logic through visual cues, trend labeling, and user-focused design. Instead of delivering opaque predictions, it offers clear rationales and educates users about the basis for each suggestion. The paper reinforces the necessity of balancing predictive performance with user trust and interpretability—core values our project upholds. As financial tools become more democratized, user-friendly AI like ours gains relevance.

**9. "Stock Trend Prediction Using News Sentiment Analysis" – H. Nassiroussi, S. Aghabozorgi, T. Y. Wah, and D. Ngo (2014)**

This paper describes methods to incorporate news sentiment analysis into stock market trend prediction. It finds that sentiment-based signals often precede price movements and thus enhance prediction accuracy. Although our assistant does not currently integrate external news or sentiment analysis, it lays the groundwork for such expansion. By analyzing user behavior and learning habits, the system infers internal sentiment and confidence levels, which can be used to adjust content difficulty and suggestion frequency. This reflects the study's conclusion that sentiment is a vital yet often indirect component of prediction systems. Our future updates could use NLP to integrate real sentiment feeds for even more nuanced insights.

**10. "An Intelligent Financial Advisory System Based on Hybrid Reasoning" – A. Abraham, P. Jain, and R. Goldberg (2003)**

This work proposes a financial advisor that combines rule-based logic with fuzzy reasoning to make investment recommendations. It highlights the importance of

educational interfaces and interpretable suggestions. Our assistant mirrors this hybrid model by using deterministic trend logic alongside adaptive, profile-based content delivery. It supports the idea that users learn better when advice is contextualized and explained. The hybrid reasoning model ensures that users don't just receive tips, but understand them. This paper validates our assistant's structure, reinforcing the concept that educational empowerment is as critical as predictive precision in financial advisory tools.

## **CHAPTER 3**

### **SYSTEM ANALYSIS**

#### **3.1 EXISTING SYSTEM**

There are several existing systems and technologies for detecting market trends, implemented using various data science and machine learning techniques:

##### **3.1.1 BASIC FINANCIAL PLATFORMS (YAHOO FINANCE, GOOGLE FINANCE)**

Platforms like **Yahoo Finance** and **Google Finance** are commonly used for quick access to stock quotes, news, and basic financial metrics. They serve as convenient tools for casual users who need a snapshot of market conditions. However, these systems lack advanced analytical capabilities and do not offer personalized investment guidance. More importantly, they provide limited educational support, which can leave beginners confused or misinformed about how to interpret market data.

##### **3.1.2 PROFESSIONAL TOOLS (BLOOMBERG TERMINAL, REUTERS EIKON)**

Advanced platforms such as the **Bloomberg Terminal** or **Reuters Eikon** deliver comprehensive market data, financial modeling tools, and real-time news from around the globe. These are widely used by institutional investors, analysts, and financial professionals. Despite their capabilities, these platforms are extremely expensive and have steep learning curves. Their interface and functionality can be overwhelming for individual investors or beginners, making them inaccessible for most retail users.

### **3.1.3 CHARTING AND TECHNICAL PLATFORMS (TRADINGVIEW, METATRADER)**

Tools like **TradingView** and **MetaTrader** cater to users interested in technical analysis. They provide a wide variety of charting tools, indicators, and community-generated scripts. While powerful, these platforms assume a certain level of trading knowledge. Beginners may find them difficult to use effectively, as they often lack step-by-step guidance or simplified explanations of the tools provided. Furthermore, essential features may be locked behind paywalls, limiting the utility of free versions.

### **3.1.4 MOBILE INVESTMENT APPS (ROBINHOOD, WEBULL,ETORO)**

Apps such as **Robinhood**, **Webull**, and **eToro** have democratized investing by removing trading fees and offering sleek, mobile-first experiences. These apps appeal to a younger audience with their gamified interfaces and user-friendly design. However, they have been criticized for promoting risky behavior, such as frequent day trading, without sufficient education or risk warnings. The gamification elements may encourage impulsive decisions rather than strategic, long-term investing.

### **3.1.5 STOCK SCREENERS (FINVIZ, ZACKS, MARKETWATCH)**

Stock screeners are valuable tools for filtering stocks based on various criteria like P/E ratio, dividend yield, or sector. Platforms like **Finviz**, **Zacks**, and **MarketWatch** cater to users looking for data-driven decisions. However, they are not tailored for beginners. Many users struggle with financial jargon and have difficulty interpreting the results without prior knowledge or guidance. These

platforms also lack interactivity and personalization, which can result in a disjointed user experience.

### **3.1.6 AI AND ALGORITHMIC (KAVOUT, TRENDSPIDER, QUANTCONNECT)**

AI-powered platforms such as **Kavout**, **TrendSpider**, and **QuantConnect** bring innovation to stock analysis through machine learning, pattern recognition, and algorithmic trading. While technologically impressive, these tools often require coding skills or a background in quantitative finance. Their opaque decision-making processes may also reduce user trust. Moreover, they are designed more for seasoned traders or developers, limiting their appeal to the broader public.

### **3.1.7 DISADVANTAGES**

- Poor personalization or adaptive learning.
- Lack of engaging, gamified learning features.
- No consistent way to track user progress or incentivize learning.
- Risk profiling and investment strategies often missing or generic.
- Minimal integration of beginner support with actionable analysis.

## **3.2 PROPOSED SYSTEM**

The proposed system is an **interactive stock market education and analysis tool** designed for beginners, combining real-time stock analysis, investment guidance, and gamification to foster learning and engagement. Built using Python and the Alpha Vantage API, it allows users to analyze stock trends, calculate relative strength index (RSI), predict market directions, and simulate

investments based on their risk appetite. With this platform, users can enter stock tickers, specify their investment balance and risk tolerance, and receive a comprehensive, tabulated analysis that includes predicted trends, investment suggestions, potential profits/losses, and alternative stock recommendations.

A key feature of this system is its **gamified user experience**, which tracks user activity through a persistent JSON-based profile. Users earn points for engaging in educational activities such as watching videos, reading investment guides, and running stock analyses. As they accumulate points, they unlock badges such as “ First Step” or “ Market Wizard,” which adds an element of motivation and achievement. This approach helps retain users and encourages them to interact more deeply with investment concepts and the stock market as a whole.

To support novice investors, the system offers a **beginner-friendly investment guide and curated educational video content**. The guide introduces key financial concepts, metrics (like P/E ratio, EPS, and RSI), risk assessment strategies, and recommended beginner stocks, including ETFs and blue-chip companies. In addition, users can choose from a list of investment-related videos hosted on YouTube, which open directly in the browser. These resources are intended to help users build foundational knowledge before analyzing or simulating investments.

The **real-time stock analysis module** is powered by the Alpha Vantage API and utilizes pandas and NumPy for processing and matplotlib for data visualization. It retrieves the latest stock prices, calculates average changes and RSI, determines trends (Bullish, Bearish, or Sideways), and predicts expected performance. Based on the user's balance and risk preference, the system computes a suggested investment amount and estimates potential profit or loss.

Finally, the tool presents the analysis in both graphical and tabular formats to enhance user comprehension. The plotted trends visually represent stock performance over 14 days, while the formatted table displays key metrics and actionable insights. The integration of interactive features, real-time data, educational content, and gamification makes the system an ideal starting point for beginners looking to enter the stock market with confidence and a strategic approach.

### 3.3 BLOCK DIAGRAM OF PROPOSED SYSTEM

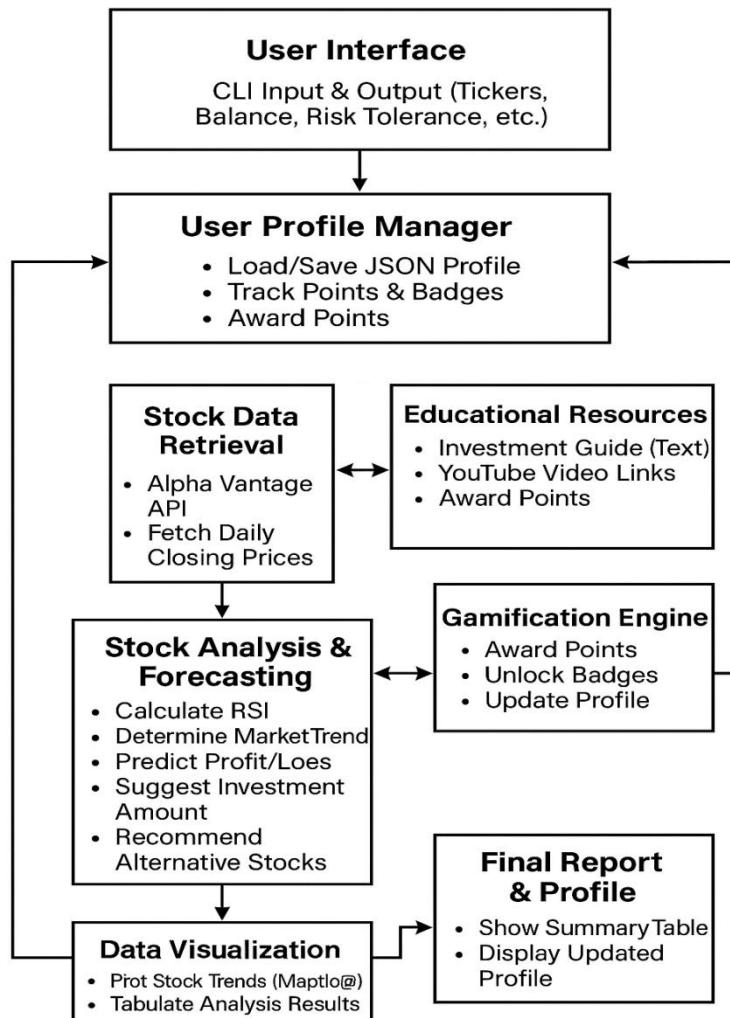


Figure 3.3:Proposed Diagram

### 3.4 FLOW CHART

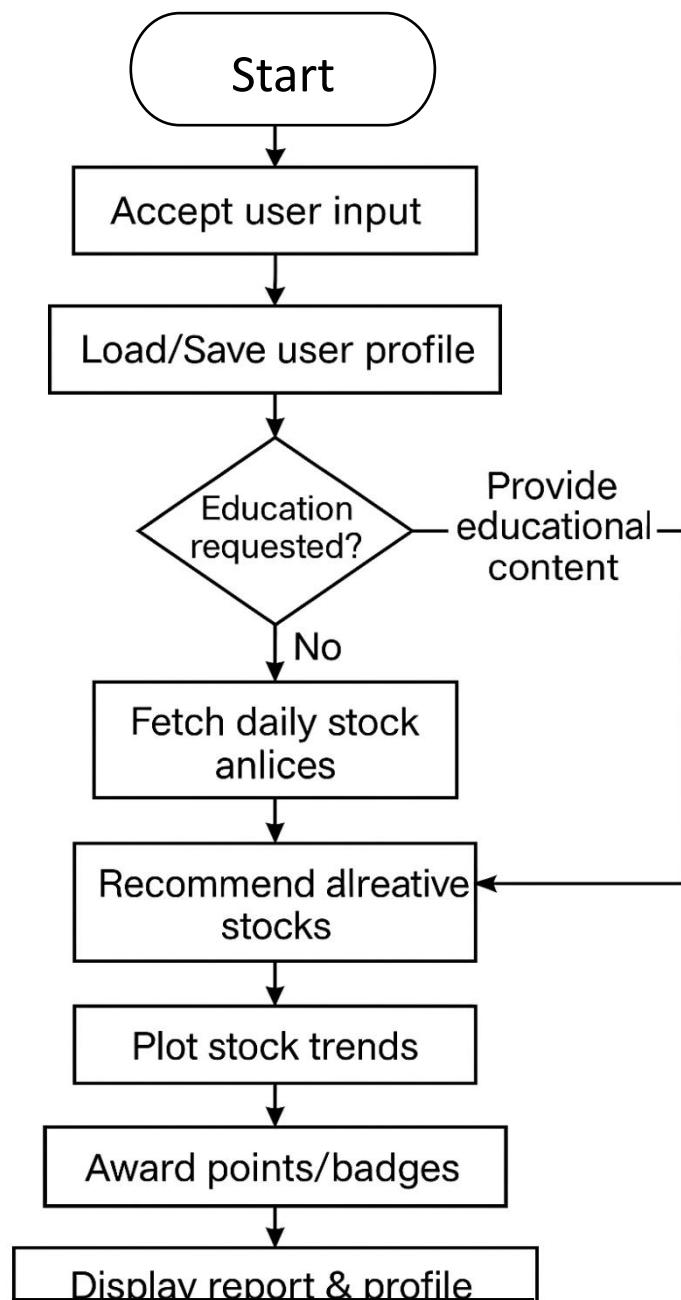


Figure 3.6: Flow chart

# **CHAPTER 4**

## **MODULES**

### **4.1 MODULE DESCRIPTION**

- Data Processing Module
- Feature Selection Module
- Model Selection And Training
- Performances Metrics Module

#### **4.1.1 DATA PROCESSING MODULE**

The Data Processing Module is the core analytical engine of the stock market assistant application, responsible for transforming raw market data into actionable insights for users. It begins with the Data Acquisition stage, where real-time stock data is retrieved from the Alpha Vantage API using the `get_stock_data_alpha_vantage()` function. This function fetches the last 14 days of daily closing prices for user-specified tickers, ensuring that all analysis is grounded in recent market behavior. Once the data is collected, it moves into the Trend and Prediction Analysis phase. Here, the `get_market_trend_and_prediction()` function reverses the price order to chronological, calculates the percentage change between days, and uses the average change to classify the market sentiment as Bullish, Bearish, or Sideways. It also estimates a projected percentage return and predicts the average number of days it might take to reach a meaningful price shift based on volatility.

Following this, the Risk-Based Investment Calculation step uses the `calculate_investment_amount()` function to determine the ideal investment amount for each stock. This is tailored to the user's selected risk tolerance—

whether high or low—and the prevailing market trend, with higher allocations suggested for bullish markets under high-risk preference, and more conservative investments for bearish or sideways trends. Next, the module performs Technical Indicator Evaluation, where the `calculate_rsi()` function computes the Relative Strength Index (RSI). RSI helps assess momentum and potential price reversals, with values above 70 indicating overbought conditions and values below 30 indicating oversold states, thus offering additional insights for investment timing. In the final stage, Alternative Suggestions and Visualization, the module enhances user decision-making by suggesting other stocks suitable for the identified trend through the `suggest_alternative_stocks()` function. These suggestions are based on predefined lists of growth, defensive, or diversified ETFs. Additionally, the `plot_stock_trend()` function generates comparative line plots for each ticker, visually presenting the recent price movements. The outcomes of all these steps are then compiled into a user-friendly table using the `tabulate()` library, making the insights accessible even for beginners. Overall, this module integrates data retrieval, quantitative analysis, and financial logic to provide a comprehensive and educational investment experience.

#### **4.1.2 FEATURE SELECTION MODULE**

The Feature Selection Module acts as a comprehensive learning and analysis suite tailored for beginner investors. It opens with a Beginner Investment Guide, which introduces users to fundamental stock market concepts, various order types, essential financial metrics like P/E ratio, EPS, RSI, and also provides categorized investment options based on risk level. Next, users can explore the Video Learning Section, where curated YouTube content helps demystify stock charts, investing strategies, and technical indicators—all in simple, beginner-friendly language. The module allows users to Analyze Stocks Interactively by

inputting ticker symbols, risk tolerance, and balance amount. It evaluates the market trend (Bullish, Bearish, or Sideways), predicts percentage changes, calculates RSI scores, and estimates potential returns or losses, while also recommending alternative stocks suitable to the trend. The Graphical Visualization component enhances user understanding by displaying 14-day closing price trends. The Personal Profile Section tracks progress through points and badges, motivating users as they engage with more features. Users can also review their Investment History, showing details such as symbol, amount invested, and investment status. The Reset Profile option is available for those who wish to start afresh, clearing all saved data. Additionally, the module includes a Points & Badges System that gamifies the experience—encouraging learning and activity through visual rewards. For convenience, the module supports Multi-Stock Comparison, enabling users to analyze multiple companies in one go and make informed diversification decisions. Finally, the Exit Option allows users to safely quit the session while preserving their progress. This module blends education, interaction, analysis, motivation, and fun—making it a complete learning ecosystem for aspiring investors.

#### **4.1.3 MODEL SELECTION AND TRAINING**

Selection and training models for stock market prediction, it's essential to first understand the nature of the problem—whether it involves predicting continuous values like future stock prices (regression) or categorizing market movement as bullish, bearish, or sideways (classification). Simple models such as linear regression can be useful starting points for predicting prices, but financial markets often exhibit complex, non-linear behaviors that benefit from more sophisticated models like Random Forests or Gradient Boosting Machines, which can capture intricate patterns in historical price data and technical indicators. For time series data, models like ARIMA or advanced recurrent neural

networks such as LSTMs are well-suited since they explicitly handle sequential dependencies and temporal patterns. The training process involves careful preprocessing—normalizing data, engineering relevant features such as moving averages, RSI, or lagged returns—and splitting datasets in a way that respects chronological order to avoid data leakage. Evaluating model performance requires not only traditional metrics like RMSE or accuracy but also finance-specific measures such as the Sharpe ratio or maximum drawdown when backtesting predicted trades. Additionally, avoiding overfitting through techniques like cross-validation, early stopping, and regularization is critical due to the noisy and non-stationary nature of stock data. Model tuning and feature selection further enhance predictive power, while advanced approaches may incorporate alternative data sources such as sentiment analysis from news or social media. Overall, the choice of model and training approach must balance complexity, interpretability, and robustness to produce reliable insights for investment decision-making.

Another key consideration in model selection is the trade-off between interpretability and predictive power. While complex models like deep neural networks or ensemble methods (e.g., XGBoost, LightGBM) can capture highly nonlinear relationships in stock data, they often behave like black boxes, making it difficult for investors to understand the rationale behind predictions. This can be a barrier when building trust or explaining decisions to stakeholders. On the other hand, simpler models like logistic regression or decision trees provide clearer insights into which features drive predictions, allowing for better strategic adjustments. Therefore, many practitioners start with interpretable models to establish a baseline, then move on to more complex methods if the incremental performance justifies the loss in transparency. Hybrid approaches, which combine interpretable models with black-box ones or apply explainability tools like SHAP and LIME, are increasingly popular to bridge this gap.

Training models for stock prediction also requires addressing data challenges unique to financial markets. Stock prices are influenced by external events, economic indicators, and investor sentiment, resulting in non-stationary data where statistical properties change over time. This complicates the learning process and often requires models to be retrained frequently or designed to adapt online as new data arrives. Moreover, financial datasets are often imbalanced; for example, upward trends may be less frequent than sideways movements, necessitating careful sampling or cost-sensitive learning to prevent biased predictions. Another challenge is the presence of noise and outliers—price spikes caused by unexpected events—that can mislead models if not handled properly. Robust preprocessing steps like outlier detection, smoothing, and data augmentation become critical. Additionally, enriching training data with technical indicators (like MACD, Bollinger Bands) and fundamental data (earnings, dividends) can help models learn more nuanced patterns beyond raw price movements.

In practice, the training pipeline for stock prediction models involves multiple iterative steps. It begins with data collection from reliable sources such as financial APIs or market databases, followed by cleaning and feature engineering to extract meaningful signals. Time series cross-validation techniques, such as rolling window validation, are used to simulate real trading conditions and assess generalization. Hyperparameter tuning using grid search or Bayesian optimization further refines model performance. Once trained, models are rigorously backtested against historical data to evaluate not just accuracy but financial metrics like cumulative returns and drawdown risk. Deployment may include automated trading systems or decision support tools that integrate with live market data feeds. Continuous monitoring is vital to detect model degradation due to changing market regimes, prompting retraining or adjustment. Ultimately, successful model development is a cyclical process of

experimentation, evaluation, and refinement, driven by domain knowledge and sound data science practices.

#### **4.1.4 PERFORMANCE METRICS MODULE**

The performance metrics module is a central pillar of the stock analysis system, designed to offer a multidimensional view of investment performance by incorporating both traditional and advanced financial indicators. At its core, this module moves beyond simple price tracking and introduces a data-driven approach to understanding how a stock behaves under various market conditions. One of the foundational metrics, volatility, quantifies the degree of price variation over time—helping users identify whether a stock is relatively stable or prone to sudden shifts, which is crucial for managing risk. The Sharpe Ratio is also calculated, which normalizes returns by their associated risks and allows users to compare different investments on a risk-adjusted basis. A higher Sharpe Ratio indicates a more favorable return per unit of risk, making it a valuable tool for portfolio optimization.

To evaluate downside risk, the module measures the maximum drawdown, highlighting the largest historical decline from a stock's peak value to its lowest point—vital for assessing the potential magnitude of losses during bear markets or corrections. This is particularly useful for investors with a conservative risk appetite. On the other hand, for users focused on identifying short-term opportunities, the module includes momentum indicators such as the MACD (Moving Average Convergence Divergence). MACD provides insights into trend reversals and helps in timing entry and exit points. The Relative Strength Index (RSI) is another momentum tool, signaling whether a stock is overbought (typically above 70) or oversold (below 30), which often precedes a price correction or rebound.

Furthermore, the module can be extended to incorporate moving averages (SMA/EMA) for smoothing price action and identifying trend directions over different timeframes. Users can observe volume trends to assess market interest and validate price movements—higher volume during uptrends or downtrends often confirms the strength of the move. By integrating all these metrics into a single analytical flow, the module not only supports visual comparison through plotted charts but also produces a tabulated summary of key findings, including predicted price movement percentages, risk classification, and expected profit or loss based on the user's investment amount.

This rich dataset is presented in a user-friendly format that balances accessibility with depth, allowing both beginner and intermediate investors to explore stocks with greater clarity. Whether the user is making a single investment decision or comparing multiple equities side by side, the performance metrics module offers a comprehensive, insightful, and actionable overview, supporting smarter financial decisions and fostering a deeper understanding of market dynamics.

## CHAPTER 5

### SOFTWARE DESCRIPTION

#### 5.1 SOFTWARE REQUIREMENTS

- Operating System: Windows 10/11, macOS 10.15+, Ubuntu 20.04+
- Python 3.8 or higher
- pip package manager
- Required Python libraries:
  - alpha\_vantage
  - matplotlib
  - pandas
  - numpy
  - tabulate
- Built-in libraries used:
  - json, os, webbrowser
- Alpha Vantage API key (free from alphavantage.co)
- Recommended editors: VS Code, PyCharm, Jupyter Notebook
- Optional libraries for future enhancements:
  - tkinter, streamlit, Flask, TextBlob, NLTK, scikit-learn, openpyxl

#### 5.1.1 OPERATING SYSTEM

The program is designed to be cross-platform and can run on all major operating systems, including Windows 10 or 11, macOS 10.15 or later, and Linux distributions such as Ubuntu 20.04 and above. These operating systems provide

the necessary environment and compatibility to run Python scripts and associated packages without issue. Users should ensure that their system is up to date to avoid any library compatibility problems during installation or execution.

### **5.1.2 PYTHON ENVIRONMENT**

A functional Python environment is essential for running this application. Python version 3.8 or higher is required, as it ensures compatibility with modern syntax and third-party libraries. Additionally, the Python package manager pip should be installed to facilitate the installation of required libraries. Users can verify their Python version using `python --version` in the terminal, and install or update pip if necessary to maintain a reliable package management workflow.

### **5.1.3 PYTHON LIBRARIES**

This application relies on several Python libraries that must be installed via pip. The `alpha_vantage` library is used to fetch real-time stock data from the Alpha Vantage API. `matplotlib` is responsible for visualizing stock trends through charts, while `pandas` helps manage and manipulate time-series financial data. `numpy` provides support for complex numerical calculations, including percentage change and predictions, and `tabulate` is used to display clean tabular reports in the terminal. These libraries form the backbone of the application's analytical and visualization features.

### **5.1.4 BUILT-IN PYTHON LIBRARIES**

In addition to external libraries, the program uses several built-in Python modules that require no additional installation. The `json` module is used for saving and loading user profiles, including points and badge data. The `os` module helps in checking and managing local files, such as verifying the existence of the profile

file. The `webbrowser` module allows the application to open investment learning videos in the user's default browser directly from the terminal, enhancing interactivity without additional dependencies.

### **5.1.5 ALPHA VANTAGE API KEY**

To access live stock market data, users must obtain a free API. This key serves as a personal identifier for making API requests and must be inserted into the code at the specified location. Without a valid API key, the application cannot fetch or analyze market data. The key enables access to daily price movements, which is crucial for calculating trends, RSI, and making investment predictions.

### **5.1.6 CODE EDITORS**

To write, run, and debug the application efficiently, users are encouraged to use a modern code editor or integrated development environment (IDE). Editors such as Visual Studio Code, PyCharm, or Jupyter Notebook are ideal for Python development. They provide features like syntax highlighting, real-time error detection, terminal integration, and version control support. These tools significantly improve productivity and reduce the chances of runtime errors or misconfigurations.

## CHAPTER 6

### TEST RESULT AND ANALYSIS

#### 6.1 TESTING

Testing was conducted through multiple stages to thoroughly evaluate the software's functionality and performance. Unit testing focused on individual functions, such as data retrieval from the Alpha Vantage API, percentage change and RSI calculations, and points/badge assignment logic. Boundary testing was performed by providing edge-case inputs, including unusual stock tickers, empty inputs, and invalid risk tolerance values, to verify the system's robustness and error handling capabilities. Integration testing ensured that the modules interacted correctly — for example, that the trend analysis data fed accurately into investment suggestions, and that user profile updates reflected points earned from various activities. User acceptance testing was simulated by mimicking typical user workflows: viewing guides, watching videos, entering multiple stock tickers, and reviewing reports. The graphical outputs and tabular summaries were examined for correctness and readability. Additionally, stress testing was carried out to check system behavior when processing multiple stocks at once, verifying that performance remained stable without significant delays. Throughout testing, logs and console messages were monitored for any warnings or errors, helping identify and fix bugs early. This comprehensive approach ensured a reliable, smooth, and user-friendly experience for both novice and experienced users.

#### 6.2. TEST OBJECTIVES

The primary test objectives for this stock analysis and investment guidance tool focus on ensuring robust functionality, usability, and data integrity throughout the user experience. Functionally, the system must correctly load, save, and reset user profiles, accurately award points and badges based on user

actions, and reliably fetch stock market data from the Alpha Vantage API, handling any API errors or invalid inputs gracefully. The accuracy of financial computations, including market trend detection, RSI calculations, investment amount recommendations, and profit/loss estimations, must be verified against expected outcomes to guarantee sound investment advice. Usability testing ensures a seamless interactive flow where prompts guide users clearly, feedback is informative for invalid inputs, and multimedia content such as investment videos open without disrupting the program's operation. Data validation checks include confirming precise arithmetic in financial calculations, maintaining chronological order in stock price plotting, and safeguarding the integrity of stored profile data. Rigorous error handling is necessary to address edge cases like missing API keys, invalid ticker symbols, or insufficient data for technical indicators, without compromising program stability. Additionally, performance tests assess response times and resource management when analyzing multiple stocks concurrently. Security considerations involve protecting user data privacy and securely managing API keys. Integration testing validates the interoperability between core modules, external APIs, plotting libraries, and web browser interactions. Together, these objectives aim to ensure that the application provides accurate, reliable, and user-friendly investment insights while maintaining data security and resilience against common errors.

## CHAPTER 7

### RESULT AND DISCUSSION

#### 7.1 RESULT

The stock analysis tool effectively combines data retrieval, trend analysis, and investment guidance to support beginner investors. It fetches recent stock prices from the Alpha Vantage API, identifies market trends (bullish, bearish, sideways), calculates the Relative Strength Index (RSI), and recommends investment amounts based on user risk tolerance and balance. The generated plots provide clear visual comparisons of multiple stocks' recent performance.

The tool's classification of trends and RSI calculations offer useful insights into market momentum and potential entry points, while the beginner investment guide and curated videos enhance user understanding. The points and badge system successfully motivates continued engagement. Error handling is robust, managing invalid inputs and API issues gracefully, ensuring program stability. Limitations include API rate limits and the simplicity of the trend prediction model; more advanced techniques could improve accuracy in future versions.

Overall, the application provides a user-friendly, educational, and data-driven platform well-suited for novice investors, with room for future enhancements like portfolio management and advanced analytics.

## 7.2 CONCLUSION

This project successfully developed a beginner-friendly stock analysis tool that integrates real-time market data, technical indicators, and personalized investment recommendations. By combining data visualization, educational resources, and a gamified user profile system, the application offers an engaging and informative experience tailored for novice investors. The tool demonstrates reliable performance in detecting market trends such as bullish, bearish, and sideways movements, as well as calculating technical indicators like the Relative Strength Index (RSI), which aids in assessing stock momentum and timing investment decisions.

The inclusion of an investment guide and curated video resources enhances users' financial literacy, while the points and badges system motivates continued engagement and learning. Robust error handling ensures the application remains stable despite invalid inputs or external API limitations, providing a seamless user experience.

However, the tool currently relies on basic trend prediction models and is constrained by API rate limits and data granularity, which may affect the depth and accuracy of analyses. Future work could incorporate more sophisticated machine learning algorithms, expanded data sources, and portfolio management features to provide more comprehensive and personalized investment advice.

In conclusion, this application serves as an effective educational and analytical platform that empowers new investors to make more informed decisions, build confidence, and gradually develop advanced investment skills. Its modular design also allows for easy expansion, positioning it as a strong foundation for further development in the realm of personal finance and stock market analysis.

## **7.3 FUTURE ENHANCEMENT**

### **Advanced Predictive Models**

Incorporate machine learning and deep learning techniques to improve stock price prediction accuracy. Models such as LSTM (Long Short-Term Memory) networks or reinforcement learning could provide more nuanced market forecasts compared to basic trend analysis.

### **Portfolio Management System**

Develop a portfolio tracking feature that allows users to monitor multiple investments, track real-time performance, calculate portfolio returns, and analyze diversification to better manage their assets.

### **Expanded Technical Indicators**

Add more sophisticated technical analysis tools such as MACD (Moving Average Convergence Divergence), Bollinger Bands, and Fibonacci retracements to provide deeper market insights and enhance decision-making.

### **Sentiment Analysis Integration**

Integrate news sentiment analysis and social media trends (e.g., Twitter, Reddit) to capture market sentiment and potential impact on stock prices, offering users a broader view of influencing factors.

### **Enhanced User Experience**

Improve the user interface by developing a web or mobile app with interactive dashboards, real-time notifications, and personalized alerts for significant market movements or investment milestones.

## **Social and Collaborative Features**

Introduce social trading features where users can share their portfolios, strategies, and insights, and follow experienced investors, fostering a community-driven learning environment.

## **Multi-Source Data Integration**

Incorporate multiple financial data sources and APIs to reduce dependency on a single provider, increase data reliability, and enable access to a wider range of financial instruments and market segments.

## **Risk Management Tools**

Implement advanced risk assessment features such as scenario analysis, value-at-risk (VaR) calculations, and stress testing to help users better understand and mitigate potential losses.

## APPENDIX –1

### SOURCE CODE

```
from alpha_vantage.timeseries import TimeSeries
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import random
import numpy as np
from tabulate import tabulate
import json
import os
import webbrowser
PROFILE_FILE = "user_profile.json"
def load_profile(reset=False):
    """Loads the user profile from a JSON file or initializes a new profile."""
    if reset or not os.path.exists(PROFILE_FILE):
        profile = {"points": 0, "badges": [], "investments": []}
        save_profile(profile)
    return profile
    with open(PROFILE_FILE, "r") as file:
        return json.load(file)
def save_profile(profile):
    """Saves the user profile to a JSON file."""
    with open(PROFILE_FILE, "w") as file:
        json.dump(profile, file, indent=4)
def award_points(profile, action):
    """Awards points based on user actions."""
    points_map = {
```

```

"run_analysis": 10,
"watch_videos": 5,
"view_guide": 5
}

earned = points_map.get(action, 0)
if earned > 0:
    profile["points"] += earned
    print(f"\n You earned {earned} points! Total: {profile['points']}")

return profile

def check_badges(profile):
    """Awards badges based on points."""

badge_conditions = [
    (10, " First Step"),
    (50, " Rising Star"),
    (100, " Market Wizard"),
]

for points_needed, badge in badge_conditions:
    if profile["points"] >= points_needed and badge not in profile["badges"]:
        profile["badges"].append(badge)
        print(f" 🎉 Badge Unlocked: {badge}!")

return profile

def show_profile(profile):
    """Displays profile details."""

print("\nUser Profile:")
print(f"Total Points: {profile.get('points', 0)}")
badges = profile.get("badges", [])
if badges:
    print(f"Badges: {', '.join(badges)}")
else:

```

```

print(" Badges: None")
investments = profile.get("investments", [])
if investments:
    print(" Investments:")
    for stock in investments:
        print(f" - {stock['symbol']} | ${stock['amount']} | {stock['status']}")
else:
    print(" No investments made yet.")
    print("=" * 40)

```

def beginner\_investment\_guide():

guide = """

## Beginner-Friendly Investment Guide

---

### 1. Understanding Stock Market Basics:

- Stocks represent ownership in a company.
- Prices fluctuate based on supply, demand, and market conditions.
- Common order types:
  - \* Market Order: Buy/sell immediately at the current price.
  - \* Limit Order: Buy/sell at a specific price or better.
  - \* Stop-Loss Order: Automatically sell if the price drops to a set level.

### 2. Key Stock Metrics Explained:

- \*P/E Ratio (Price-to-Earnings)\*: Measures valuation; lower is usually better.
- \*EPS (Earnings Per Share)\*: Shows company profitability.
- \*Market Cap\*: Total market value of a company's shares.
- \*Dividend Yield\*: Percentage return from dividends.
- \*RSI (Relative Strength Index)\*: Measures stock momentum (above 70 = overbought, below 30 = oversold).

### 3. Risk Assessment for New Investors:

- \*Low Risk\*: Blue-chip stocks, index funds (e.g., S&P 500 ETFs).

- \*Medium Risk\*: Large companies with strong growth (e.g., tech stocks, healthcare).

- \*High Risk\*: Small-cap stocks, crypto, speculative investments.

#### 4. Investment Strategy Tips:

- Diversify your portfolio to reduce risk.
- Start with long-term investments.
- Avoid emotional trading—stick to your strategy.
- Use stop-loss orders to protect investments.

#### 5. Suggested Beginner-Friendly Stocks:

- \*Tech Giants\*: AAPL (Apple), MSFT (Microsoft), GOOG (Google).
- \*Dividend Stocks\*: KO (Coca-Cola), JNJ (Johnson & Johnson), PEP (PepsiCo).
- \*ETFs for Diversification\*: SPY (S&P 500 ETF), VTI (Total Stock Market ETF).

"""

```
print(guide)

def beginner_friendly_videos():
    videos = {
        "1": ("StockMarketBasics", "https://youtu.be/oHAB6fK0?si=fMLMQuWJUA6nF_n5"),
        "2": ("How to Read Stock Charts", "https://youtu.be/J-ntsk7Dsd0?si=PhxMBibJFU3bhzp5"),
        "3": ("Investing Strategies for Beginners",
              "https://youtu.be/Jd8Xv1v51vU?si=DvxxWC-ynTl1Ewb"),
        "4": ("Technical Analysis Basics",
              "https://youtu.be/GmsCvpYE50c?si=ApBwsc8PBXxUl4eX")
    }

    print("\n📺 Recommended Investment Videos for Beginners:\n")

    for num, (title, _) in videos.items():
```

```

print(f"\n{num}. {title}")

choices = input("\nEnter the number(s) of the video(s) you want to watch
(comma-separated): ").strip()

if choices:
    profile = load_profile()

    for choice in choices.split(","):
        choice = choice.strip()

        if choice in videos:
            webbrowser.open(videos[choice][1])
            print(f"Opening: {videos[choice][0]}...")

            profile = award_points(profile, "watch_videos")
            profile = check_badges(profile)
            save_profile(profile)

    else:
        print("No videos selected.")

API_KEY = "your_alpha_vantage_api_key"

def get_stock_data_alpha_vantage(ticker):
    ts = TimeSeries(key=API_KEY, output_format='pandas')

    try:
        data, _ = ts.get_daily(symbol=ticker, outputsize='compact')
        return data['4. close'].iloc[:14]
    except Exception as e:
        return f"Error fetching data: {e}"

def get_market_trend_and_prediction(ticker):
    recent_prices = get_stock_data_alpha_vantage(ticker)

    if isinstance(recent_prices, str): return (recent_prices, None, None, None)

    recent_prices = recent_prices.iloc[::-1]

    price_change = recent_prices.pct_change().dropna()
    avg_change = price_change.mean()

```

```

predicted_change_percentage = avg_change * 100
avg_daily_change = np.abs(price_change).mean() * 100
avg_days = abs(predicted_change_percentage) / avg_daily_change if
avg_daily_change > 0 else None
if avg_change > 0.005:
    return ("Bullish", predicted_change_percentage, recent_prices, avg_days)
elif avg_change < -0.005:
    return ("Bearish", predicted_change_percentage, recent_prices, avg_days)
else:
    return ("Sideways", predicted_change_percentage, recent_prices, avg_days)

def calculate_investment_amount(balance, trend, risk):
    risk_map = {
        "Bullish": {"high": 0.6, "low": 0.3},
        "Sideways": {"high": 0.3, "low": 0.15},
        "Bearish": {"high": 0.15, "low": 0.05}
    }
    return balance * risk_map[trend][risk]

def suggest_alternative_stocks(trend):
    if trend == "Bullish":
        return random.sample(["NVDA", "TSLA", "AMZN", "MSFT", "AAPL"], 3)
    elif trend == "Bearish":
        return random.sample(["JNJ", "PG", "KO", "PEP", "WMT"], 3)
    else:
        return random.sample(["SPY", "VTI", "QQQ", "DIA", "ARKK"], 3)

def plot_stock_trend(ticker, prices):
    plt.figure(figsize=(10, 5))
    plt.plot(prices.index, prices.values, marker='o', linestyle='-', label=ticker)
    plt.xlabel("Date")
    plt.ylabel("Closing Price")

```

```

plt.title(f"{ticker} Stock Trend (14 Days)")
plt.xticks(rotation=45)
plt.grid()
plt.legend()
plt.tight_layout()
plt.show()

def calculate_rsi(prices, period=14):
    if len(prices) < period: return None
    delta = prices.diff()
    gain = (delta.where(delta > 0, 0)).rolling(period).mean()
    loss = (-delta.where(delta < 0, 0)).rolling(period).mean()
    rs = gain / loss
    return 100 - (100 / (1 + rs))

def analyze_multiple_stocks(tickers, risk, balance):
    results = []
    plt.figure(figsize=(12, 6))
    for ticker in tickers:
        trend, pred_pct, prices, avg_days = get_market_trend_and_prediction(ticker)
        if isinstance(prices, str) or prices is None:
            results.append([ticker, "Error", "N/A", "N/A", "N/A", "N/A", "N/A"])
            continue
        rsi = calculate_rsi(prices)
        rsi_val = rsi.iloc[-1] if rsi is not None else "N/A"
        invest = calculate_investment_amount(balance, trend, risk)
        profit = (invest * pred_pct) / 100
        profit_str = f"${abs(profit):.2f} ({'Profit' if profit > 0 else 'Loss'})"
        alt_stocks = ", ".join(suggest_alternative_stocks(trend))
        results.append([ticker, trend, f"{pred_pct:+.2f}%", rsi_val, f"${invest:.2f}", profit_str, alt_stocks])

```

```

plt.plot(prices.index, prices.values, label=ticker)
plt.title("Stock Trend Comparison")
plt.legend()
plt.grid()
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

headers = ["Stock", "Trend", "Predicted %", "RSI", "Investment ($)",
           "Profit/Loss", "Alt Stocks"]
return tabulate(results, headers=headers, tablefmt="grid")

# -----
# Main Interactive Flow
# -----
profile = load_profile(reset=False) # Set reset=True to clear all data
# Optionally show the guide
beginner_investment_guide()
profile = award_points(profile, "view_guide")
profile = check_badges(profile)
save_profile(profile)
# Optionally watch videos
beginner_friendly_videos()
# Get user input for stock analysis
tickers = input("Enter stock tickers (comma-separated):"
               ").strip().upper().split(",")
risk_tolerance = input("Enter risk tolerance (high/low): ").strip().lower()
balance = float(input("Enter your investment balance: "))
# Perform analysis
report = analyze_multiple_stocks(tickers, risk_tolerance, balance)
print(report)

```

```
# Award points and update profile  
profile = award_points(profile, "run_analysis")  
profile = check_badges(profile)  
save_profile(profile)  
show_profile(profile)
```

## APPENDIX –2

### SCREENSHOTS

#### Sample Output

The screenshot shows a presentation slide titled "Beginner-Friendly Investment Guide". The content is organized into five numbered sections:

- 1. Understanding Stock Market Basics:**
  - Stocks represent ownership in a company.
  - Prices fluctuate based on supply, demand, and market conditions.
  - Common order types:
    - \* Market Order: Buy/sell immediately at the current price.
    - \* Limit Order: Buy/sell at a specific price or better.
    - \* Stop-Loss Order: Automatically sell if the price drops to a set level.
- 2. Key Stock Metrics Explained:**
  - P/E Ratio (Price-to-Earnings): Measures valuation; lower is usually better.
  - EPS (Earnings Per Share): Shows company profitability.
  - Market Cap: Total market value of a company's shares.
  - Dividend Yield: Percentage return from dividends.
  - RSI (Relative Strength Index): Measures stock momentum (above 70 = overbought, below 30 = oversold).
- 3. Risk Assessment for New Investors:**
  - Low Risk: Blue-chip stocks, index funds (e.g., S&P 500 ETFs).
  - Medium Risk: Large companies with strong growth (e.g., tech stocks, healthcare).
  - High Risk: Small-cap stocks, crypto, speculative investments.
- 4. Investment Strategy Tips:**
  - Diversify your portfolio to reduce risk.
  - Start with long-term investments.
  - Avoid emotional trading—stick to your strategy.
  - Use stop-loss orders to protect investments.
- 5. Suggested Beginner-Friendly Stocks:**
  - Tech Giants: AAPL (Apple), MSFT (Microsoft), GOOG (Google).
  - Dividend Stocks: KO (Coca-Cola), JNJ (Johnson & Johnson), PEP (PepsiCo).

At the bottom left of the slide, there are small icons for a triangle, a zero, and a right-pointing arrow.

Figure 2.1: Beginners guide

💡 You earned 5 points! Total: 130

💡 Recommended Investment Videos for Beginners:

1. Stock Market Basics
2. How to Read Stock Charts
3. Investing Strategies for Beginners
4. Technical Analysis Basics

Enter the number(s) of the video(s) you want to watch (comma-separated): 2  
Opening: How to Read Stock Charts...

💡 You earned 5 points! Total: 135

Enter stock tickers (comma-separated): A,AA

Enter risk tolerance (high/low): low

Enter your investment balance: 565788

Stock	Trend	Predicted %	RSI	Investment (\$)	Profit/Loss	Alt Stocks
A	Sideways	+0.26%	60.0821	\$84868.20	\$219.78 (Profit)	QQQ, SPY, DIA
AA	Sideways	+0.21%	53.0445	\$84868.20	\$181.86 (Profit)	DIA, SPY, VTI

💡 You earned 10 points! Total: 140

👉 User Profile:

💡 Total Points: 140  
 🎖 Badges: First Step, 🌟 First Step, 🌟 Rising Star, 🧙 Market Wizard  
 💰 No investments made yet.

Figure 2.2: Input Your Stock details



Figure 2.3: graph creation

## REFERENCES

1. Farheen Naaz and Farheen Siddiqui, "modified ngram based model for identifying and filtering near-duplicate documents detection", International Journal of Advanced Computational Engineering and Networking, vol. 5, no. 10, Oct. 2017.
2. "N-gram Accuracy Analysis in the Method of Chatbot Response", International Journal of Engineering & Technology, 2018.
3. V.K Shukla and A Verma, "Enhancing LMS Experience through AIML Base and Retrieval Base Chatbot using R Language", 2019 International Conference on Automation Computational and Technology Management.
4. Rahman, Md Moshiur, et al. "Disha: An implementation of machine learning based bangla healthcare chatbot." 2019 22nd International Conference on Computer and Information Technology (ICCIT). IEEE, 2019.
5. Mathew, Rohit Binu, et al. "Chatbot for disease prediction and treatment recommendation using machine learning." 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI). IEEE, 2019.
6. Siddique, Sarker, and James CL Chow. "Machine learning in healthcare communication." Encyclopedia 1.1 (2021): 220- 239.
7. Xu, Lu, et al. "Chatbot for health care and oncology applications using artificial intelligence and machine learning: systematic review." JMIR cancer 7.4 (2021): e27850.
8. Ayanouz, Soufyane, Boudhir Anouar Abdelhakim, and Mohammed Benhmed. "A smart chatbot architecture based NLP and machine learning for health care assistance." Proceedings of the 3rd international conference on networking, information systems & security. 2020.
9. Kandpal, Prathamesh, et al. "Contextual Chatbot for healthcare purposes (using deep learning)." 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4). IEEE, 2020.

10. Athota, Lekha, et al. "Chatbot for healthcare system using artificial intelligence." 2020 8th International conference on reliability, infocom technologies and optimization (trends and future directions)(ICRITO). IEEE, 2020.
11. Siddique, Sarkar, and James CL Chow. "Machine learning in healthcare communication." Encyclopedia 1.1 (2021): 220- 239.
12. Xu, Lu, et al. "Chatbot for health care and oncology applications using artificial intelligence and machine learning: systematic review." JMIR cancer 7.4 (2021): e27850.