A PROJECT REPORT

On

Algorithm Trading Bot

##### Submitted in partial fulfillment of the requirement of University of Mumbai for the Degree of

**Bachelor of Engineering**

In

**CSE AIML**

Submitted By

**Miss. Riya C Jadhav**

**Mr. Tejas B Kamble**

**Miss. Gaurangi C Raul**

**Miss. Kshitija D Satpute**

Supervisor

**Prof. Vandana Shinde**



Department of CSE-Artificial Intelligence and Machine Learning Smt. Indira Gandhi College of Engineering, Ghansoli – 400701 UNIVERSITY OF MUMBAI

#### Academic Year 2023 – 24



Department of CSE-Artificial Intelligence and Machine Learning

SMT. INDIRA GANDHI COLLEGE OF ENGINEERING

GHANSOLI – 400701

## CERTIFICATE

This is to certify that the requirements for the Major Project entitled ‘**Algorithm Trading Bot**’ have been successfully completed by the following students:

Name Roll No.

Miss. Riya Jadhav 25

Mr. Tejas Kamble 30

Miss. Gaurangi Raul 54

Miss. Kshitija Satpute

in partial fulfillment of Bachelor of Technology of Mumbai University in the Department of CSE Artificial Intelligence and Machine Learning, SMT. INDIRA GANDHI COLLEGE OF ENGINEERING, GHANSOLI – 400701 during the Academic Year 2023 – 2024.

**Supervisor**

**Prof. Vandana Shinde**

**Head of Department Principal**

**Prof. Sonali Deshpande Dr. Sunil Chavan**



Department of CSE-Artificial Intelligence and Machine Learning

SMT. INDIRA GANDHI COLLEGE OF ENGINEERING

GHANSOLI – 400701

## PROJECT APPROVAL FOR B.E

This project entitled **“****Algorithm Trading Bot”** by **Riya Jadhav, Tejas Kamble, Gaurangi Raul, Kshitija Satpute** Name are approved for the degree of **CSE-Artificial Intelligence and Machine Learning**

Examiners:

1.

2.

Supervisors:

1.

2.

Chairman:

1.

Date:

Place:



Department of CSE-Artificial Intelligence and Machine Learning

SMT. INDIRA GANDHI COLLEGE OF ENGINEERING

GHANSOLI – 400701

## DECLARATION

We declare that this written submission for the B.E project entitled “**Algorithm Trading Bot**” represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any ideas / data / fact / source in our submission. We understand that any violation of the above will cause disciplinary action by the institute and also evoke penal action from the sources which have not been properly cited or from whom prior permission have not been taken when needed.

Project Group Members:

Riya Jadhav:

Tejas Kamble: Gaurangi Raul: Kshitija Satpute:

Date:

Place

# Table of Contents

|  |  |
| --- | --- |
| Abstract................................................................................................................................ | i |
| List of Figures...................................................................................................................... | ii |
| **1.** Introduction................................................................................................................ | 1 |
| **1.1** Fundamentals................................................................................................... | 2 |
| **1.2** Objectives........................................................................................................ | 3 |
| **1.3** Organization of the Project Report….............................................................. | 4 |
| **2.** Literature Survey....................................................................................................... | 5 |
| **3** Project Overview................................................................................... | 13 |
| **4** Methodology…...................................................................... | 14 |
| **5** Project Design & Process workflow............................................................................ | 19 |

Abstract

Title: - Algorithm Trading bot

A trading bot using the Kotak API is a software program that can automatically place and execute trades on the Kotak Securities trading platform. The bot can be programmed to use a variety of trading strategies, such as technical analysis, fundamental analysis, or a combination of both.

The emergence of the market for derivatives products, most notably forwards, futures and options, can be traced back to the willingness of risk-averse economic agents to guard themselves against uncertainties arising out of fluctuations in asset prices.

The Kotak API provides developers with access to a wide range of data and functionality, including:Live market data, Historical market data, Account information, Order placement and execution, Trading strategy backtesting and optimization.

Once deployed, the trading bot will be able to automatically place and execute trades on the Kotak Securities trading platform according to the developer's trading strategy.

Algorithmic trading uses algorithms that follow a trend and defined set of instructions to perform a trade. The trade can generate revenue at an inhuman and enhanced speed and frequency. The characterized sets of trading guidelines that are passed on to the program are reliant upon timing, value, amount, or any mathematical model. Aside from profitable openings for the trader, algo-trading renders the market more liquid and trading more precise by precluding the effect of human feelings on trading. Our project aims to further this revolution in the markets of tomorrow by providing an effective and efficient solution to overcome the drawbacks faced due to manual trading by building an Algorithmic Trading Bot which will automatically trade user strategies alongside its own algorithms for day-to-day trading based on different market conditions and user approach ,and throughout the course of the day invest and trade with continuous modifications to ensure the best trade turnover for the day while reducing the transaction cost, hence enabling huge profits for concerned users be it Organizations or individuals.

**List of Figures**

|  |  |
| --- | --- |
| Fig. 1.2 | 3 |
| Fig 2.2 | 5 |
| Fig 3.1 | 7 |
| Fig 3.2 | 10 |
| Fig 3.3 | 12 |

**Chapter 1**

###### INTRODUCTION

A trading bot is a software program that can automatically buy and sell financial assets on behalf of a human trader. Trading bots can be used to trade a variety of assets, including stocks, bonds, currencies, and cryptocurrencies.

Trading bots are typically programmed with a set of trading rules, such as when to buy and sell assets based on technical indicators or market conditions. Once programmed, trading bots can execute trades automatically without any human intervention.Trading bots can be used by traders of all experience levels, from beginners to experts. They can be a valuable tool for traders who want to:Automate their trading strategies: Trading bots can save traders a lot of time and effort by automating the trading process. This can be especially beneficial for traders who trade multiple markets or who have complex trading strategies.

* 1. **Fundamentals**

Market Analysis and Strategy Development: Before creating a trading bot, it's essential to develop a clear understanding of the financial markets and the strategies you want to implement. This involves analyzing historical data, identifying patterns, and developing algorithms that exploit those patterns.

Programming Skills: Developing an automated trading bot typically requires proficiency in programming languages such as Python, Java, or C++. These languages are commonly used for algorithmic trading due to their versatility and extensive libraries for data analysis and trading platform integration.

API Access: Most trading platforms provide APIs (Application Programming Interfaces) that allow developers to connect their algorithms to the platform for executing trades. Understanding and effectively utilizing these APIs is crucial for building a functional trading bot.

Risk Management: Implementing robust risk management techniques is essential to mitigate the potential losses associated with automated trading. This includes setting stop-loss orders, position sizing, and incorporating risk-reward ratios into trading strategies.

Backtesting: Before deploying a trading bot in live markets, it's crucial to test its performance using historical data. Backtesting allows you to evaluate the effectiveness of your trading strategy and make necessary adjustments before risking real capital.

Real-Time Data Processing: Automated trading bots need to process real-time market data efficiently to identify trading opportunities and execute trades promptly. Implementing mechanisms for data streaming and processing is essential for maintaining the bot's responsiveness.

Monitoring and Maintenance: Continuous monitoring and maintenance are necessary to ensure that the trading bot operates effectively in changing market conditions. This includes monitoring performance metrics, debugging potential issues, and updating trading strategies as needed.

Regulatory Compliance: Depending on your jurisdiction and the type of assets you're trading, there may be regulatory requirements that govern automated trading activities. It's essential to understand and comply with relevant regulations to avoid legal issues.

Security: Security is paramount when dealing with automated trading bots, as they often involve handling sensitive financial data and executing trades on behalf of users. Implementing robust security measures to protect user data and prevent unauthorized access is critical.

Emotional Discipline: One of the advantages of automated trading is the elimination of emotional bias from trading decisions. However, it's important to ensure that the trading algorithms are designed to maintain discipline and avoid irrational decision-making.

* 1. **Objectives**

**Profit Maximization**: The primary objective of an automated trading bot is often to generate profits by executing trades that capitalize on market opportunities identified by the trading algorithm.

**Risk Management**: Implementing effective risk management strategies to minimize potential losses is crucial. Objectives related to risk management include setting stop-loss orders, diversifying portfolios, and adhering to predetermined risk limits.

**Consistency**: Achieving consistency in trading performance is essential for long-term success. Objectives related to consistency may include maintaining a stable return on investment (ROI) over time and minimizing drawdowns.

**Efficiency**: Maximizing the efficiency of trading operations is key to optimizing returns. Objectives related to efficiency include minimizing transaction costs, reducing latency in trade execution, and optimizing order routing.

**Adaptability**: Financial markets are dynamic and subject to constant change. An automated trading bot should be adaptable to evolving market conditions. Objectives related to adaptability include incorporating machine learning techniques to adapt trading strategies and parameters in response to market dynamics.

**Scalability**: As the trading strategy proves successful, there may be objectives related to scaling up trading operations. This could involve increasing the size of trades or expanding into new markets while maintaining profitability and risk management standards.

**Transparency**: Providing transparency to users or stakeholders about the bot's operations and performance is essential for building trust. Objectives related to transparency may include providing detailed performance reports, disclosing trading strategies, and maintaining clear communication channels with users.

**Compliance**: Ensuring compliance with relevant regulations and legal requirements is paramount. Objectives related to compliance may include adhering to trading regulations, maintaining data privacy standards, and implementing robust security measures.

Continuous Improvement: Striving for continuous improvement is essential to stay competitive in financial markets. Objectives related to continuous improvement may include conducting regular performance reviews, optimizing trading algorithms based on backtesting results, and staying informed about advancements in trading technology.

User Satisfaction: If the automated trading bot is used by clients or customers, objectives related to user satisfaction are crucial. This could involve providing a user-friendly interface, offering responsive customer support, and delivering satisfactory returns on investment.

* 1. Organization of the Report

Introduction: The report outlines the development process of an automated trading bot, highlighting its significance in financial markets and setting the stage for subsequent discussions.

Market Analysis and Strategy Development: Detailed analysis of financial markets and the formulation of trading strategies based on algorithmic approaches are discussed, emphasizing the importance of strategy development in achieving trading objectives.

Technical Development: The report delves into the technical aspects of bot development, including programming languages, tools, and APIs used, as well as data processing techniques employed for efficient operation.

Risk Management: Integration of robust risk management strategies, such as stop-loss orders and position sizing, is emphasized to mitigate potential losses and ensure the bot's stability in varying market conditions.

Backtesting and Performance Evaluation: Methodologies for backtesting trading strategies and evaluating performance metrics are presented, highlighting the importance of thorough testing before deployment in live markets.

Real-Time Operation: Mechanisms for real-time data processing, trade execution, and monitoring are explained, showcasing the bot's ability to respond promptly to market changes and execute trades efficiently.

Regulatory Compliance and Security: Considerations for regulatory compliance and implementation of security measures are outlined, underlining the importance of adhering to legal requirements and safeguarding user data.

Results and Discussion: A summary of achieved objectives, challenges encountered, and lessons learned from the development process is provided, offering insights into the bot's effectiveness and areas for improvement.

Conclusion and Recommendations: The summary concludes by summarizing key findings and insights, along with recommendations for further enhancement and future research, ensuring continuous improvement in bot performance and functionality.

# Chapter 2

# Literature Survey

Title: Literature Survey on Automated Trading Bot Development and Implementation

Objective: This literature survey provides an overview of existing research and literature related to the development and implementation of automated trading bots in financial markets, offering insights into best practices, challenges, and advancements in the field.

Key Points:

Introduction to Automated Trading: Literature in this area introduces the concept of automated trading, its evolution, and its significance in modern financial markets. Key topics include the automation of trading strategies, algorithmic trading techniques, and the role of technology in market efficiency.

Algorithmic Trading Strategies: Research explores various algorithmic trading strategies employed by automated trading bots, including trend-following strategies, mean reversion strategies, statistical arbitrage, and machine learning-based approaches. Studies analyze the effectiveness of these strategies in different market conditions and asset classes.

Technical Development and Implementation: This section discusses the technical aspects of developing and implementing automated trading bots, including programming languages, software libraries, and APIs commonly used in bot development. Research highlights best practices for system architecture design, data processing, and trade execution mechanisms.

Risk Management and Performance Evaluation: Literature examines risk management techniques integrated into automated trading bots to mitigate potential losses and optimize risk-adjusted returns. Studies also discuss methodologies for backtesting trading strategies and evaluating performance metrics, emphasizing the importance of robust risk management and thorough performance analysis.

Real-Time Operation and Market Dynamics: Research in this area explores the challenges and opportunities associated with real-time operation of automated trading bots in dynamic financial markets. Topics include data streaming and processing, order execution algorithms, and strategies for adapting to changing market conditions.

Regulatory Compliance and Security: Studies address regulatory considerations and security challenges associated with automated trading, including compliance with trading regulations, data privacy standards, and cybersecurity measures. Research offers insights into best practices for ensuring regulatory compliance and safeguarding user data in automated trading systems.

Case Studies and Empirical Analysis: Literature includes case studies and empirical analysis of automated trading bots in real-world scenarios, examining their performance, profitability, and impact on market dynamics. Studies analyze factors influencing bot performance, such as market liquidity, transaction costs, and technological infrastructure.

Challenges and Future Directions: This section discusses challenges and limitations of automated trading bot development and implementation, such as overfitting, data snooping bias, and market manipulation risks. Research also identifies emerging trends and future directions in automated trading, including advancements in artificial intelligence, blockchain technology, and decentralized finance (DeFi)

### Chapter 3

###### PROJECT SCOPE

###### Scope Statement:

###### Define the boundaries and objectives of the project, including: Developing software for automated trading based on predefined algorithms. Integrating the trading bot with selected trading platforms or APIs.Implementing risk management strategies to mitigate potential losses.Conducting backtesting to evaluate the bot's performance. Providing documentation and support for users.

###### Key Features and Functionality: Specify the key features and functionality to be included in the automated trading bot, such as:

###### Algorithmic trading strategies (e.g., trend-following, mean reversion).

###### Real-time data processing and analysis.

###### Trade execution and order management.

###### Risk management mechanisms (e.g., stop-loss orders, position sizing).

###### Performance monitoring and reporting.

###### Target Markets

###### and Assets:

###### Define the target markets (e.g., equities, forex, cryptocurrencies) and asset classes for trading.

###### Specify any restrictions or preferences regarding the types of assets to be traded.

###### Technology Stack:Identify the technology stack required for bot development, including programming languages, development frameworks, and third-party libraries or APIs. Specify any hardware or software requirements for running the trading bot efficiently.

###### Regulatory Compliance:Address regulatory considerations relevant to automated trading in the target markets. Ensure compliance with applicable laws, regulations, and industry standards governing automated trading activities.

###### Project Constraints:

###### Timeframe: Define the project timeline, including start and end dates, as well as key milestones and deadlines.

###### Budget: Specify the project budget, including resources required for development, testing, and deployment.

###### Resources: Identify the project team members and their roles, as well as any external resources or expertise needed.

###### Risk Management:

###### Identify potential risks and uncertainties that may impact project success, such as technical challenges, market volatility, or regulatory changes. Develop strategies to mitigate and manage identified risks throughout the project lifecycle.

###### Testing and Quality Assurance: Outline the testing methodologies and quality assurance processes to be implemented, including: Unit testing of individual components. Integration testing of system modules.User acceptance testing (UAT) with simulated or historical market data.

###### Documentation and Training:Develop comprehensive documentation for the automated trading bot, including:

###### Technical specifications.

###### User guides and manuals.

###### Troubleshooting and support resources.

###### Provide training and support for users to ensure effective utilization of the trading bot.

### Chapter 4

**METHODOLOGY**

#### To develop a trading bot using the Kotak API

* + - 1. **Overview**

Obtain API credentials from Kotak Securities. You can do this by creating an account and navigating to the API section of your account dashboard.

Choose a programming language and development environment. The Kotak API is supported by a variety of programming languages, including Python, Java, and C++.

Install the Kotak API client library for your chosen programming language. The Kotak API client library provides a set of classes and functions that make it easy to interact with the Kotak API.

Develop your trading strategy. Your trading strategy should define the rules that your trading bot will use to buy and sell assets. You can use technical indicators, fundamental analysis, or a combination of both to develop your trading strategy.

Implement your trading strategy using the Kotak API client library. The Kotak API client library provides a variety of methods that you can use to place and manage orders, retrieve market data, and get account information.

Test and optimize your trading strategy. Once you have implemented your trading strategy, you should test it on historical market data to see how it performs. You can also use optimization techniques to improve the performance of your trading strategy.

Deploy your trading bot to a live trading environment. Once you are satisfied with the performance of your trading strategy, you can deploy your trading bot to a live trading environment.

Here is a more detailed overview of each step:

1. Obtain API credentials from Kotak Securities

To obtain API credentials from Kotak Securities, you will need to create an account and navigate to the API section of your account dashboard. Once you are in the API section, you can generate a new API key and secret.

2. Choose a programming language and development environment

The Kotak API is supported by a variety of programming languages, including Python, Java, and C++. You can choose any programming language that you are comfortable with.

Once you have chosen a programming language, you will need to set up a development environment. A development environment is a software package that provides you with the tools you need to develop and run your trading bot.

3. Install the Kotak API client library for your chosen programming language

The Kotak API client library is a set of classes and functions that make it easy to interact with the Kotak API. You can download the Kotak API client library from the Kotak Securities website.

Once you have downloaded the Kotak API client library, you will need to install it on your computer. The installation process will vary depending on your programming language and development environment.

4. Develop your trading strategy

Your trading strategy should define the rules that your trading bot will use to buy and sell assets. You can use technical indicators, fundamental analysis, or a combination of both to develop your trading strategy.

Here are some examples of trading strategies:

Moving average crossover strategy: This strategy buys an asset when its price crosses above its moving average and sells it when its price crosses below its moving average.

MACD crossover strategy: This strategy buys an asset when the MACD line crosses above the signal line and sells it when the MACD line crosses below the signal line.

Relative Strength Index (RSI) strategy: This strategy buys an asset when the RSI falls below 30 and sells it when the RSI rises above 70.

5. Implement your trading strategy using the Kotak API client library

The Kotak API client library provides a variety of methods that you can use to place and manage orders, retrieve market data, and get account information.

Here is an example of how to place a buy order using the Kotak API client library:

Python

import kotak\_api

client=kotak\_api.Client(api\_key='YOUR\_API\_KEY', api\_secret='YOUR\_API\_SECRET')

order = client.place\_order(

symbol='NIFTY',

quantity=1,

side='BUY',

order\_type='MARKET'

)

print(order)

6. Test and optimize your trading strategy

Once you have implemented your trading strategy, you should test it on historical market data to see how it performs. You can use a backtesting platform to test your trading strategy on historical market data.

Backtesting will help you to identify any weaknesses in your trading strategy and to make necessary adjustments.

Once you are satisfied with the performance of your trading strategy, you can deploy it to a live trading environment.

**Features and Advantages of The RSI Indicator**

As any other oscillator, the RSI indicator is not plotted on the price chart, but in a separate window below. This technical instrument consists of a single line and two levels set by default. Vertical axis range of the indicator is set to 1 to 100 showing extremality of current price against its previous values.

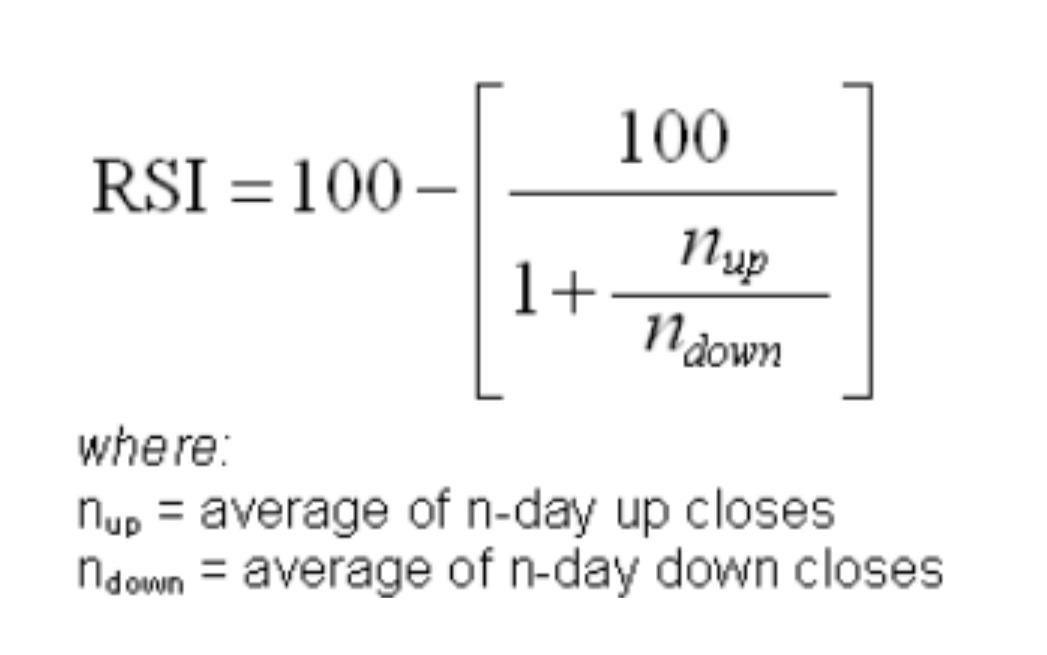
RSI calculation

RSI = 100 – (100 / (1 + U/D))

Where:

U – average number of positive price changes

D – average number of negative price changes

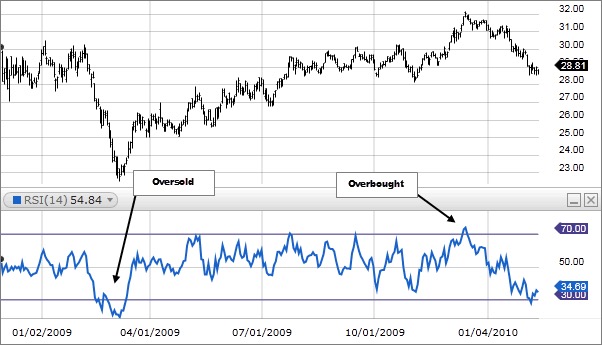


**Setting and Adjusting RSI Indicator**

There are two ways to set up this indicator. The easiest way is to click the tab ‘List of Indicators’ located on the upper panel of the terminal and select ‘Oscillators’ – ‘Relative Strength Index’.

Another option is to choose ‘Insert’ – ‘Indicators’ – ‘Oscillators’ – ‘Relative Strength Index’.

Instrument configuration window will open before the indicator is set in the chart. This window allows you to configure the indicators parameters. The main parameter is the period; It defines the number of price values taken into consideration at plotting the main indicator’s line. The shorter the period, the steeper indicator’s chart movements will be. This parameter is set to 14 by default, and this setting is considered optimal in most cases. You can also adjust the style settings, like line colour and weight. By using another tab of configuration window, you can change parameters of the levels from 30 and 70 to 20 and 80. You can also add new levels should your strategy require so.



### Chapter 5

###### PROJECT DESIGN & PROCESS WORKFLOW

* 1. **Use** Case Diagram / Activity Diagram /DFD

**5.2 Software Requirement**

1. Operating System: Windows 10/11

2. Python

3. Jupyter

4. Kotak API.

5. Trading account.

6. Neo Api Manager

Kotak Securities Trading Software (Kotak Securities Trading Platforms)

Kotak Securities offers multiple trading platforms to suit investors' requirements. Following are the trading platforms available by Kotak Sec:

1. Website based Trading

A stock trading website, which can be accessed through any popular Web Browser. Kotak Neo can be accessed on your desktop/laptop. All exciting features be it Stocks & IPO's, Futures & Options, Mutual Fund & ETFs, Currency & Commodity, Smallcase, Stockit, Pay off analyser, you will find everything under one roof.

2. Mobile Stock Trading

Kotak Neo App is designed to provide simplified and smooth trading & investing experience. The app is loaded with trader and investor-centric features, keeping in mind the needs of new age traders & investors.

3. Trade API

Through Neo Trade API, you can execute trades with utmost ease. They help traders and developers to integrate live market feed with their strategies. Customise your trading platform to plan and execute trades. The company's Fintech Partners are: Tradetron, Algo Baba, Trading View, Amibroker, Ninja Trader, Chartlink, Meta Trader 4 and Trade from Excel.

4. Desktop

Kotak Securities have Nest Terminal, a user-friendly platform that will help dynamic traders monitor markets, place orders seamlessly, and also buy & sell shares in real-time. The sophisticated trading terminal will give you an edge in the competitive market.

Kotak Securities Pros (Advantages)

The following are the advantages of Kotak Securities. You must read Kotak Securities advantages and disadvantages before opening an account with Kotak Securities. Kotak Securities pros and cons help you find if it suits your investment needs.

1. Kotak Securities offers a 3-in-1 account.

2. 25+ years of trusted service.

3. Low-cost online discount brokerage plans are available.

4. Zero brokerage plan for investors up to the age of 30 years.

10 Features of Kotak Securities Mobile App

Analyze, track and trade stocks on the go with Kotak Mobile Trading app. Key feature of the app include:

1. Online Trading on Mobile Phones, Tablets and iPAD

Equity Cash Trading (Delivery)

Equity Derivatives Trading (F&O)

Currency Derivatives Trading (F&O)

2. Track orders placed earlier

3. Track portfolio in real time

4. Live streaming quotes & charts

5. View Demat Account Holdings

6. Online Fund Transfer

7. Reports & Historic Actions

8. Live Market Info and Market Watch

9. Intraday Tick by Tick Charts

10. Manager Account / Profile

Requirements For Making bots.

To create a trading bot like the one we’re building, you’ll need the following:

Python: Python provides a versatile and powerful programming language for developing trading bots due to its extensive libraries and ease of use.

Broker API: We’re using the Kotak Neo API provided by Kotak Neo Broker for accessing market data and executing trades. However, since each user may have a different broker, they’ll need to refer to their broker’s API documentation and make necessary adjustments to the code accordingly.

WebSocket Data: Continuous and real-time market data is crucial for making timely trading decisions. Utilizing websocket data allows our bot to receive updates on the symbols we’re trading, enabling us to execute buy and sell orders promptly.

Technical Analysis Library: A library for technical analysis such as TA-Lib and pandas-ta in Python can be helpful for calculating indicators like the Relative Strength Index (RSI) used in our trading strategy.

Risk Management: Implementing proper risk management techniques is essential for protecting capital and ensuring long-term profitability. This includes setting stop-loss orders, position sizing based on account size and risk tolerance, and monitoring leverage.

Backtesting Framework: Before deploying our bot in live trading, it’s crucial to backtest our strategy using historical data thoroughly. Utilizing a backtesting framework such as Backtrader or Zipline can help simulate trading performance and identify potential flaws in the strategy. You can use a custom framework also as we used in our Backtesting Rsi trading strategy using Python.

Security Measures: Since trading bots involve automated execution of trades, ensuring the security of API keys and sensitive information is paramount. Implementing secure practices such as storing API keys securely and utilizing encrypted communication channels is essential to protect against unauthorized access.

Here’s how the function works:

It calculates the end date as the current date and time, and then the start date as one day before the end date.

The function then calls a hypothetical historical\_data() function to retrieve historical price data for the specified symbol, start date, end date, and interval. This data is stored in a DataFrame (df).

The columns of the DataFrame are renamed to match the expected format (‘Date’, ‘Open’, ‘High’, ‘Low’, ‘Close’, ‘Volume’).

Using the ta library, the function calculates the RSI (Relative Strength Index) and ATR (Average True Range) based on the closing prices.

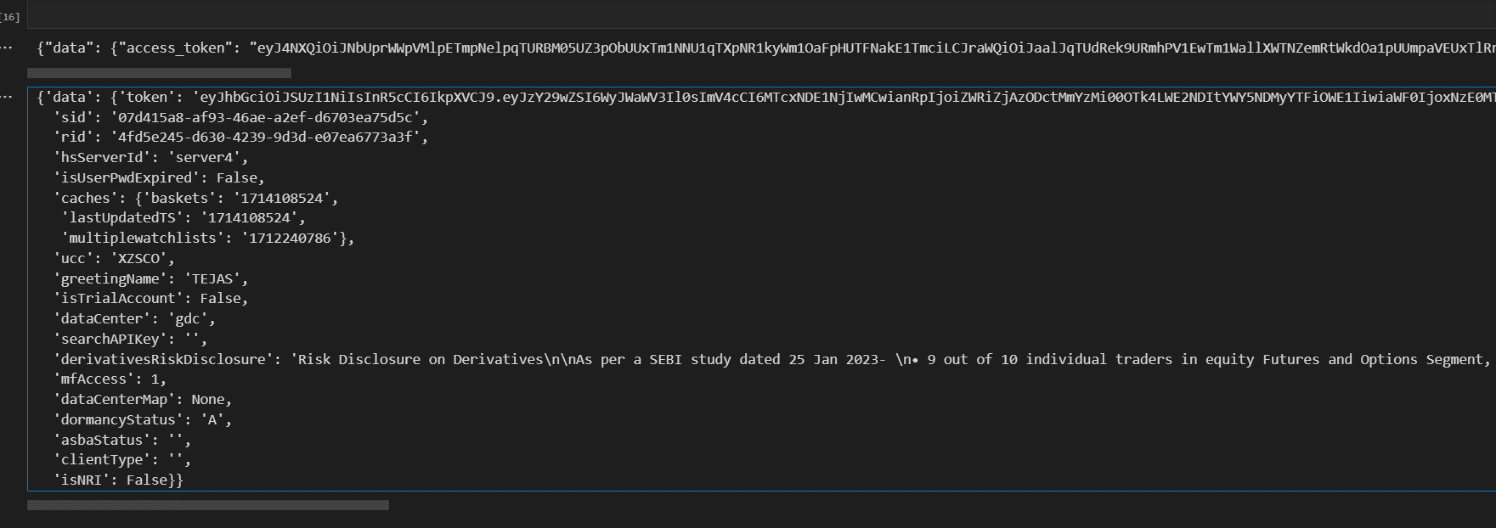
It initializes a new column called ‘crossover\_signal’ in the DataFrame and sets all values to 0.

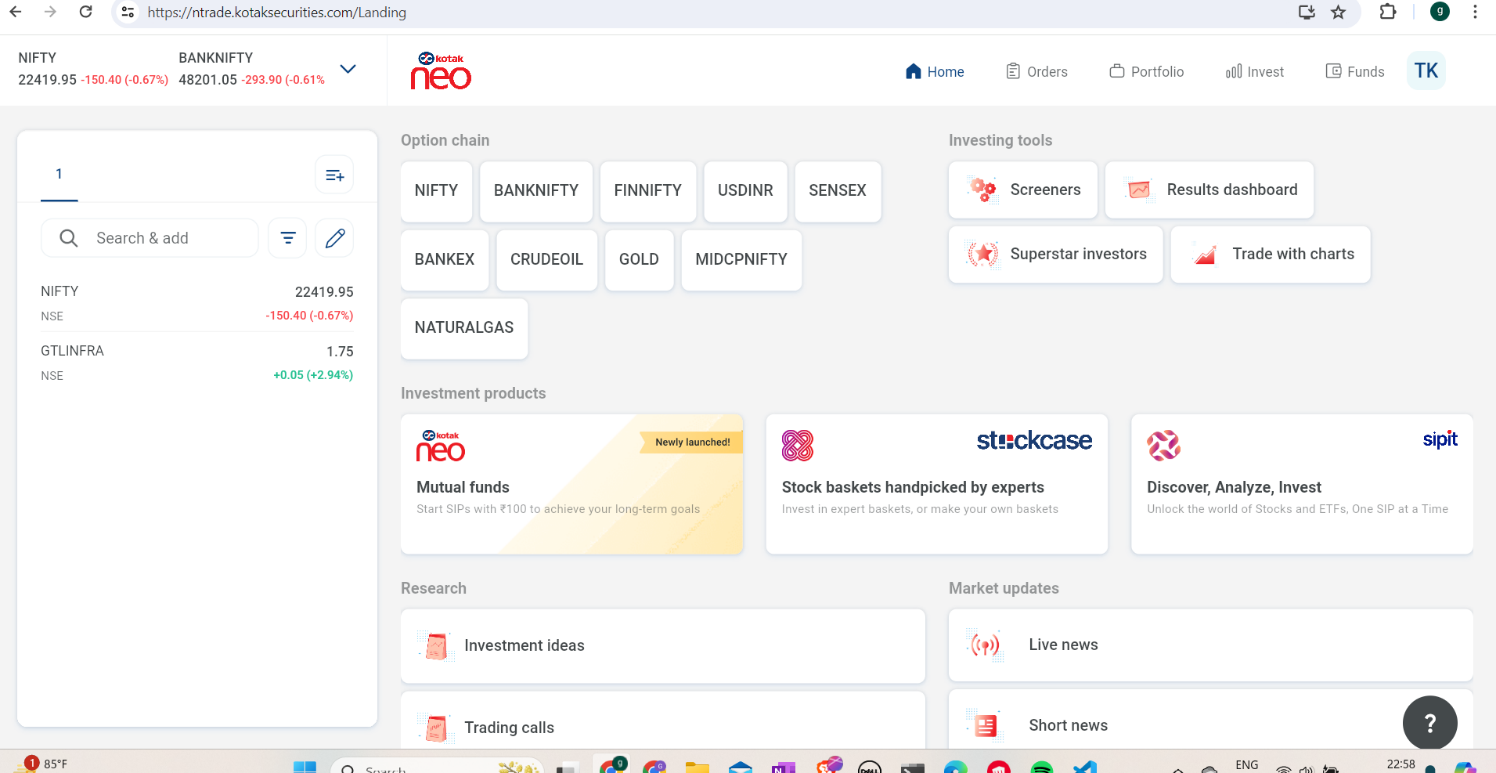
The function identifies points where the RSI crosses above or below the threshold of 50 and sets the corresponding ‘crossover\_signal’ values to 1 or -1, respectively, indicating buy or sell signals.

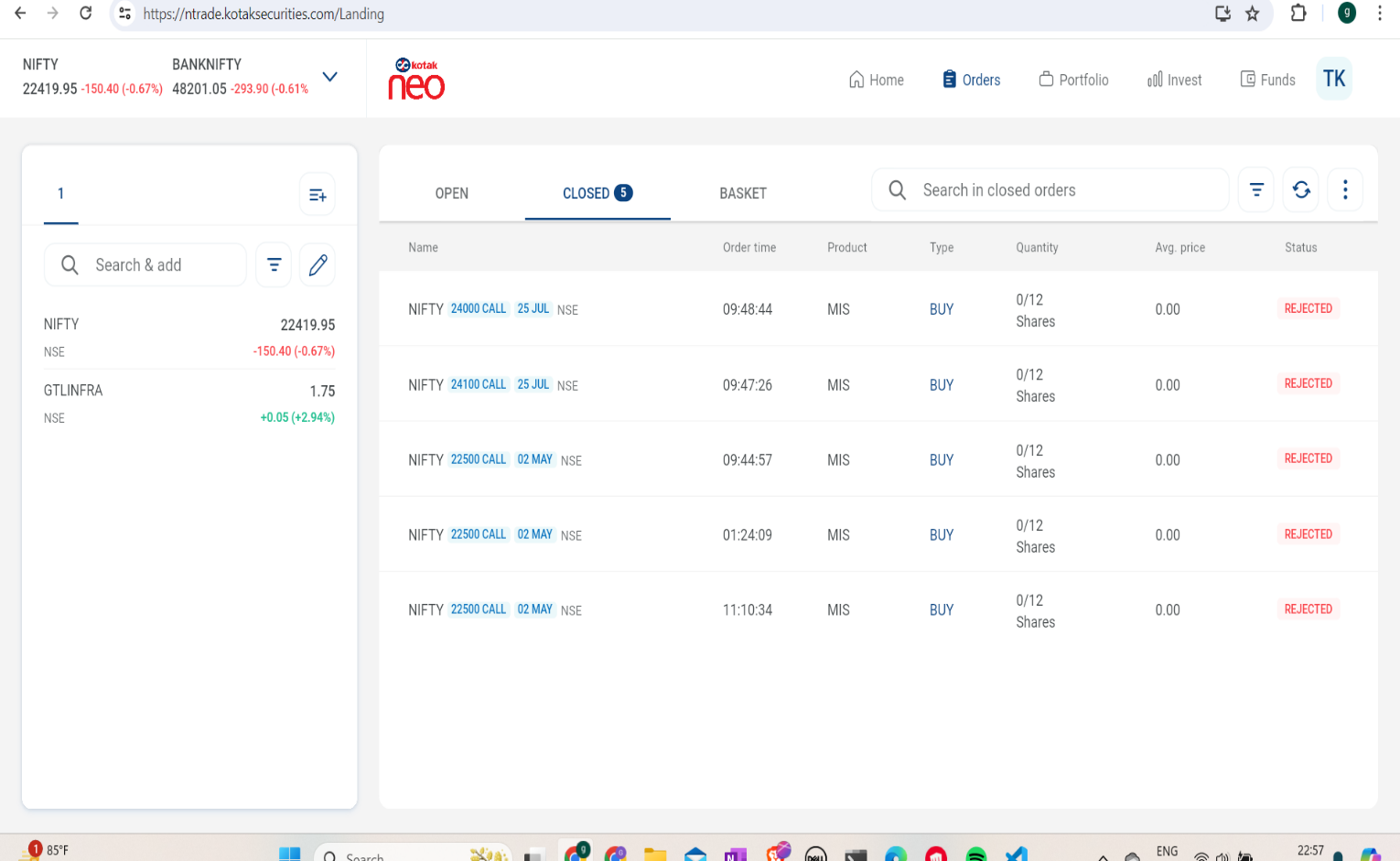
Finally, it converts the ‘Date’ column to the desired format and returns the DataFrame containing the historical data along with the RSI signals.

### Chapter 6

###### RESULTS AND APPLICATIONS







### Chapter 7

###### CONCLUSION AND FUTURE SCOPE

Conclusions:

Developing a trading bot using the Kotak API can be a challenging but rewarding experience. Trading bots can help traders to automate the trading process, save time, and make more profitable trading decisions.

However, it is important to note that trading bots are not a guaranteed way to make money. Traders should always backtest and optimize their trading strategies thoroughly before deploying them to a live trading environment. Additionally, traders should always monitor their trading bots closely and be prepared to manually intervene if necessary.

Algorithmic trading Bot not only provides Security, Cost, and Speed but is also a revolutionary technology for the future financial markets and economy. Algorithmic Trading Bot makes it easier for both new traders as well as established ones in getting profitable outcomes with minimized effort, time and loss.The integration of Financial Knowledge with Machine Learning is a demand of future Trading and enhances both Performance and Revenue.

Future scope:

The future scope of automated trading bots is quite promising, as they continue to evolve alongside advancements in technology and financial markets. Here are some aspects contributing to their future scope:

Advanced Algorithms: With advancements in machine learning and AI, automated trading bots will become more sophisticated in analyzing vast amounts of data and identifying complex trading patterns. This can lead to improved trading strategies and better decision-making.

Big Data and Predictive Analytics: As big data analytics techniques improve, automated trading bots will be able to process and analyze larger datasets in real-time, allowing for more accurate predictions and quicker reactions to market changes.

High-Frequency Trading (HFT): Automated trading bots are well-suited for high-frequency trading strategies due to their ability to execute trades at lightning speeds. As technology continues to advance, HFT strategies will become more prevalent, and automated bots will play a crucial role in executing these strategies efficiently.

Integration with Blockchain and Cryptocurrency Markets: As cryptocurrencies and blockchain technology gain more widespread adoption, automated trading bots will likely play a significant role in trading these assets. Integration with decentralized exchanges and smart contract platforms could further expand the scope of automated trading in this space.

Regulatory Considerations: As automated trading becomes more prevalent, regulators may introduce new regulations to ensure fair and orderly markets. Automated trading bots will need to adapt to these regulatory changes, which could impact their future scope and development.

Retail Trading: While automated trading has traditionally been more common among institutional investors, it is increasingly becoming accessible to retail traders through user-friendly platforms and APIs. This trend is likely to continue, leading to a broader adoption of automated trading bots among individual investors.

Risk Management and Compliance: Future automated trading bots may incorporate more advanced risk management and compliance features to help traders mitigate risks and ensure compliance with regulatory requirements.

Personalization and Customization: As technology improves, automated trading bots may offer more personalized and customizable features, allowing traders to tailor their strategies to their specific preferences and risk tolerance.

Overall, the future scope of automated trading bots looks promising, with continued advancements in technology and increasing adoption across various markets and investor segments. However, it's essential to remain mindful of potential risks and challenges, such as regulatory changes and technological limitations, as the landscape continues to evolve

### REFERENCES:

Reference

Acknowledgement

As every project is ever complete with the guidance of experts. So, we would like to take this opportunity to thank all those individuals who have contributed in visualizing this project.

We express our deepest gratitude to our project guide Prof. Vandana Shinde (CSE (AIML) Department, Smt. Indira Gandhi College of Engineering, University of Mumbai) for her valuable guidance, moral support and devotion bestowed on us throughout our work.

We would also take this opportunity to thank our project coordinator Prof. Sarita Khedikar for her guidance in selecting this project and also for providing us all the details on proper presentation of this project.

We extend our sincere appreciation to our entire professors from Smt. Indira Gandhi College of Engineering for their valuable inside and tip during the designing the project. Their contributions have been valuable in many ways that we find it difficult to acknowledge them individually.

We are also grateful to our HOD Prof. Sonali Deshpande for extending his help directly and indirectly through various channels in our project.

If I can say in words I must at the outset my intimacy for receipt of affectionate care to Smt. Indira Gandhi College of Engineering for providing such a simulating atmosphere and wonderful work environment.

We would like to express our gratitude to our College, Ramrao Adik Institute of technology our Mentor, Dr. Sangita Choudhary who have provided us with the opportunity to work on this project and given us support with guidance to make this project a success. We would also like to thank our teammates for their contribution and continued support and zeal towards this project.