Major Research Project Report on

Share Market Advisor Lite App

Submitted by:

Nikhil Kumar Vashist

Roll no: 23/DMBA/79

Submitted to:

Dr. Chandan Sharma

Assistant Professor

Delhi Technological University, Delhi

DELHI SCHOOL OF MANAGEMENT

Delhi Technological University

MAY 2025

CERTIFICATE

This is to certify that the Major Research Project titled "**Share Market Advisor Lite App**" is submitted by **Nikhil Kumar Vashist**, **2K23/DMBA/79** to Delhi School of Management, Delhi Technological University in partial fulfilment of the requirement for the award of the degree of Masters in Business Administration during the academic year 2024-25.

Signature of Faculty Guide

Dr. Chandan Sharma

Assistant Professor

Delhi School of Management

Delhi Technological University

DECLARATION

I, **Nikhil Kumar Vashist**, hereby declare that the Major Research Project Report entitled "**Share Market Advisor Lite App**" submitted to Delhi Technological University is a record of my original work submitted to **Dr. Chandan Sharma**, Assistant Professor, Delhi Technological University, Delhi. This project report is submitted in partial fulfilment of the requirements for the award of the degree of Masters in Business Administration during the academic year 2024-25.

I also declare that this project report has not been submitted to any other university or institute for the award of any degree or diploma.

Nikhil Kumar Vashist

2K23/DMBA/79

Date: 30th April 2025

ACKNOWLEDGEMENT

I am deeply indebted to **Dr. Chandan Sharma**, Assistant Professor, Delhi Technological University, Delhi for providing me with the opportunity to do my Major Research Project at Delhi Technological University, Delhi. I would also like to thank all the other faculty members of the Delhi School of Management at Delhi Technological University for their help and support.

I would also like to thank my family and friends for their support and encouragement throughout my project.

Finally, I would like to thank all the other people who helped me in any way during my project.

Nikhil Kumar Vashist

2K23/DMBA/79

Date: 30th April 2025

EXECUTIVE SUMMARY

In today's dynamic financial age, especially in the Indian share market, common investors encounter many challenges in making timely and informed decisions. The dynamic and volatile nature of the market and a plethora of financial information from different sources have a tendency to confuse common investors and make them uncertain. To address this great need for simple and easy-to-understand guidance, the "Share Market Advisor Lite" project was created as a data-driven solution to help common investors.

This project constructs an active, interactive website dashboard with Python and Flask. It utilizes actual market data from Yahoo Finance. The dashboard acts as a personal finance assistant, enabling users to view present market trends, examine the best means of allocating their portfolios, track increases in various sectors, and gain insight into the impact of foreign institutional investment. Through transforming raw data into meaningful information, the platform makes users better aware of intricate market information.

These are at the very center of the system:

Four principal analytical modules

- Market State Detection: A base layer that identifies the current market mood—bullish or bearish—based on the 100-day moving average of the Nifty 50 index. This analysis provides the foundation for all the following investment suggestions by correlating strategies with market conditions.
- Portfolio Optimization: According to the Markowitz Efficient Frontier model, the dashboard helps users optimize portfolios that yield the best returns for a specified level of risk. The software can adjust to market conditions by selecting stocks depending on the performance of different sectors and their risk-return profiles, offering a combination of investments and stability.
- Foreign Investment Analysis: The site tracks net 90-day inflows from Foreign Institutional Investors (FIIs) and Foreign Portfolio Investors (FPIs) and displays these as trends and market sentiment indicators, providing retail investors with a sense of the broader macroeconomic drivers on Indian markets.

- Sector Growth Monitoring: By monitoring how different sectors are performing, the system shows which sectors are growing substantially. This feature helps users determine where money is going and where sectors may lead to market growth in the future.

The dashboard was created through a straightforward process of four steps: data gathering, model creation, visual creation, and web deployment. By integrating these components into a user-friendly interface, the project demonstrates expertise in web development and data science as well as its utility through real-world applications.

For example, April 2025 market data analysis indicated a highly bullish pattern. Using the optimization function of the dashboard, a virtual portfolio was developed that achieved an estimated return of around 13.49%, surpassing the general retail investment return. Further, the tool identified high-growth industries like Telecommunication and Capital Goods, supporting its ability to inform strategic investment choices.

The present version targets the Nifty 50 index, but it is quite simple to extend the system since it is modular. The future versions can introduce support for other indices (such as Nifty Next 50 and Sensex), machine learning-based forecasts, news sentiment analysis, and overlays of macroeconomic data. These options would further increase the utility of the tool for various investors and scenarios.

In short, the "Share Market Advisor Lite" project combines theory with practice. It offers a current and easy solution to a big problem for retail investors. It makes complicated financial data easy to understand and allows investors to make more informed and confident decisions based on facts.

TABLE OF CONTENTS

CONTENT	PAGE NO.
Introduction	1
Literature Review	5
Methodology & Steps Taken	10
Case Study – Current Market State	12
Personal & Professional Learning	20
Conclusion	21
References	22
Annexures – Project Code	23

1. INTRODUCTION

1.1 Background of Project

With increasingly volatile and complex global financial markets, particularly the stock market, there is an urgent need to possess data-driven systems that inform investors to make wise choices. The Indian stock market, dominated by indices such as Nifty 50, is influenced by a broad array of big-picture and small-picture drivers. It is difficult for novice investors and even seasoned traders to track all the drivers that influence the market and make the best investment decisions.

The demand for smart advisory systems that can represent market states in concise form, recommend best portfolios, and choose important areas for investment is greater than ever before. In response to this demand, this project is proposed to create a dynamic interactive Share Market Advisory Dashboard using Python (Flask) and live financial data acquired from Yahoo Finance.

This project applies the practices of data science to develop graph and analysis software such as:

- Watching the market using moving averages
- Efficient Portfolio Creation with Markowitz Model
- Sectoral Growth Analysis
- FII/FPI Investment Trends

1.2 Problem Statement

Most investors experience issues like:

- Understanding what the market is like right now (up, down, etc.)
- Identifying best stocks to invest
- Reducing risk while maintaining acceptable returns
- Choosing top-performing industries for best allocation

This project brings solutions through the creation of a dashboard that gives:

- Tracking market trends using technical indicators.
- Market state-based portfolio optimization
- FII/FPI investment trend analysis
- Sectoral performance data for concentrated investing

Hence, the project is not just a demonstration of technical and analytical skills but also an attempt towards the development of tools that foster retail investor education and experience.

1.3 Objectives of the Project

The overall objectives of this project are:

- To create an advisory system for retail investors based on real-time information.
- To determine the current market condition through moving average analysis of Nifty50 data.
- To optimize portfolio allocation based on market condition under the Markowitz Efficient Frontier model.
- To provide a graphical illustration of Foreign Institutional Investors (FIIs) or Foreign Portfolio Investors (FPIs) activities.
- To emphasize the sectors that dominate yearly growth, thereby assisting investors in minimizing diversification and concentrating their investments for improved returns.

1.4 Scope of the Project

The scope of the project is:

- Information was gathered utilizing Yahoo Finance through Python.
- Classification of market state through analysis of the Nifty 50 index.
- Comparison of 50 companies listed in the index on PE ratios and sector classification.
- Quantitative optimization model-based portfolio formation.
- Interactive web-based dashboard using Flask and Chart.js.
- Applicable to Indian stock market investors, primarily retail players.

This framework is limited to the Nifty 50 index and its components, intentionally excluding macroeconomic indicators, global market factors, and sentiment indicators. Nonetheless, it presents a solid technical foundation that can be built upon in future work.

2. LITERATURE REVIEW

Abstract - *Financial Analysis Dashboard Application for Stock Exchange Listed Companies*

The paper "Financial Analysis Dashboard Application for Stock Exchange Listed Companies" by Florina Livia Covaci and others discusses the development of a web application to analyze the financials of stock exchange listed companies. The concept behind this application is to make it simple and convenient to see the financial status of companies and how their stocks are performing, in a user-friendly interface. This application is extremely useful to financial analysts, investors, traders, and people who are new to the field of finance because it presents a comprehensive view of a company's status using both fundamental and technical analysis approaches.

The app enables users to perform in-depth financial analysis by just looking up a company. After a company is chosen, the app shows key financial facts about solvency, liquidity, cash conversion cycle, bankruptcy risk, financial position, and performance. These facts are computed from public financial information and recent news that captures market changes in the last 24 hours. This combination of real-time news and structured financial facts allows users to make faster and improved decisions. In addition to analysis, the app also enables users to set up and track simulated transactions to see how imaginary investments would perform.

The authors highlight the use of technical and fundamental analysis in making investment decisions. The literature suggests that most investors apply technical analysis (66% of the studies surveyed). The paper, however, asserts that investment decisions for the long term need a better understanding of a company's financial fundamentals. The platform created fills the gap by combining both methods. It uses technical tools like Simple Moving Average (SMA), Exponential Moving Average (EMA), Relative Strength Index (RSI), and volume trends, along with important insights from income statements, balance sheets, and other important financial reports. This helps users evaluate a company's value, financial health, and market trends from different angles.

The application design follows a three-tier architecture. The model includes the presentation layer, application logic, and data layer. Angular is utilized to develop the front-end, which is responsive and dynamic. ASP.NET Core Web API is utilized on the server-side, and Entity Framework Core is utilized for data management using a Microsoft SQL Server database. The system follows the Model-View-Controller (MVC) architecture, which is scalable, modifiable, and easy to implement. The client and server communicate using JSON-formatted HTTP requests. Swagger was utilized to test and document the backend APIs.

Security and authentication are two major features of the application. ASP.NET Core Identity assists in user account management, such as password hashing, token authentication, and role-based access. The application also provides the facility of users performing certain actions such as recording and deleting transactions, which are stored and handled securely via secure token validation.

The dashboard has powerful data visualization functionality. It utilizes Apache ECharts to create handsome and interactive charts, which help users better understand complex data. The platform also uses jsPDF to export dashboard data and financial reports into downloadable PDF files, enabling users to save records or share reports easily with others.

The article's implementation section describes how functionalities such as user registration, login, transaction recording, and financial data retrieval have been implemented utilizing Angular services and HTTP requests to backend APIs. Data for analysis purposes are retrieved with the help of the Yahoo Finance API, and important performance indicators such as ROE, ROA, ROS, liquidity ratios, solvency scores, and the cash conversion cycle are computed and displayed on the dashboard.

The article finishes by highlighting how useful and important the created app is in the current day's fast-changing financial markets. It notes that while the app is beneficial in providing insights, it needs current financial data in order to operate at its maximum potential—i.e., it needs at least four years of financial data for each company in order to conduct a full analysis. To overcome this, extensions in the future are proposed such as using a more powerful financial data API, including community features for live discussion among investors, allowing live or practice trading connections with brokerage

sites, including support for powerful technical indicators like Japanese candlesticks, and allowing users more choice to customize the way data is displayed.

Briefly speaking, this financial analysis dashboard is a critical move toward making advanced financial tools accessible to all. It links basic financial information to valuable investment data, allowing both novice and veteran investors to make smarter, more informed choices.

Abstract - ***Visualizing Real-Time Data: Designing a Visual Analytics Tool for the Stock Market***

The thesis "Visualizing Real-Time Data: Designing a Visual Analytics Tool for the Stock Market" by Camilla Johansson and Rebecca Nilsson, Chalmers University of Technology and the University of Gothenburg, Sweden (2009):

This thesis considers the difficulty and opportunity of displaying vast volumes of real-time stock market information using visualization tools. The authors seek to reorient TIBCO Spotfire—a software developed for static information—to effectively work with real-time financial data. This research rests on the proposition that, even as digital technology has simplified it to obtain plenty of data, the complexity and quantity of the data, particularly in the stock market, are more than enough to overwhelm the users. It thus demands intelligent visualization techniques capable of enhancing insight, aiding decisions, and enabling users to quickly and efficiently read streaming data.

The study was carried out in collaboration with TIBCO Software AB, Spotfire Division, and was based on an iterative interaction design process. The process involved literature reviews, user interviews, and several iterations of prototyping and evaluation, both low- and high-fidelity prototypes. Asset managers, stock brokers, and analysts were the main user groups whose information needs in real-time were the focus of the design. The study was conducted in the Swedish stock market, with actors located in Gothenburg, and this provided a reasonable and realistic scope while being respectful of the fact that many of the findings could be translated internationally.

The majority of the studies contrasted static data and real-time data processing. The key finding was that several of the same methods of displaying and manipulating static data

can be applied to real-time data, as long as the system is able to rapidly adapt the visuals without confusing the user. The authors emphasized that real-time data must be fast, consistent, and easy to comprehend, with the ability to follow trends over time. These characteristics were of the highest concern to the stock market professionals interviewed, as their work involves rapid comprehension of evolving data streams.

To meet these requirements, the team created and tested a working prototype in Adobe Flash. The prototype contained three main views: market overview, portfolio view, and stock view, each of which could handle multiple levels of granularity of financial data. Multiple display media were incorporated into the tool, including tree maps, line charts, scatter plots, bar charts, and sparklines. These visualizations were selected by the authors because of their potential to expose trends, patterns, and relationships in data, particularly in limited screen spaces common to dashboards. The authors emphasized the application of Gestalt principles, pre-attentive visual properties, and user-centered design to enable intuitive interfaces for rapid decision-making.

The program had features which allowed users to interact with data. These were warnings, fast-forward and rewind time controls, data duplication, minimums and maximums tracking, and event markers. These allowed users to see important changes, compare different sets of data, and take a look back to see how things used to work. Users were also able to get additional information through filtering and hovering, so they did not have to clutter the home dashboard to do so. Design guidelines from testing stressed the program should be straightforward, clear, consistent, and flexible—so users could use the interface and customize it as they preferred, to analyze the data.

The authors analyzed software and sources of information that financial professionals employ, such as Reuters First and SIX Trader, and publicly available websites such as Google Finance. These applications utilize tables and static charts predominantly and provide limited real-time interaction. The new system, on the other hand, sought to include dynamic visualizations that are specifically linked to the real-time stock performance, to overcome the latencies and lack of flexibility in existing systems.

In the final chapters, the authors introduced a complete conceptual model of their system, including visual design guidelines, layout schematics, and navigation strategies. They

suggested an enhanced, good-looking interface with modular views, flexible data interaction, and sophisticated alert systems. Notably, they defined general and domain-specific design principles for real-time data visualization, which included reducing cognitive load, indicating abnormal data points, employing the right display media for data type, and excluding non-essential decorative styles that could divert attention away from data interpretation.

Lastly, the thesis also recognizes the computational requirements and technical constraints of real-time data visualization, and further research is necessary to ensure performance and scalability. Future development points suggested by the authors are live streaming integration, user customization, and cross-platform deployments, with the emergence of mobile and cloud-based analytics platforms in mind.

This study is an innovative attempt to combine interaction design and financial data visualization. It offers a solid approach to convert static traditional dashboards into dynamic, real-time decision-making tools, especially in the high-pressure stock market. The suggested improvements to TIBCO Spotfire, combined with careful user experience design, are a worthwhile contribution to information visualization and fintech technology.

3. METHODOLOGY & STEPS TAKEN

The approach that has been developed for this specific project is thoughtfully designed and formulated based on the basic principles of data collection, processing, analysis, and visualization. In the following sections, the following steps will be described in detail to outline the whole workflow of which we will be adhering to:

Step 1: Requirement Gathering & Concept Design

- Specified user requirements: retail investor decision-making assistance.
- Selected four analytical components: market trend, optimal portfolio, FII data, sectoral growth.

Step 2: The Process of Choosing the Right Technology Stack

- Backend: Python Flask
- Frontend: HTML, CSS, JavaScript, Chart.js
- Data Source: Yahoo Finance using yfinance library
- Analytics: NumPy, Pandas, PyPortfolioOpt

Step 3: How to Gather Data

- I downloaded and retrieved the historical closing prices of the Nifty 50 index and specific stocks using the yfinance library.
- Extracted PE ratios from ticker info objects.
- Developed CSV data files for Nifty50 stocks.

Step 4: Performing a Detailed Market State Analysis

- Used 100-day moving average to categorize market into:
    - o Bullish
    - o Pullback
    - o Correction
    - o Bearish
- Presented the results in a graphically appealing line chart format.

Step 5: Portfolio Optimization Process

- Applied the Markowitz Model using historical returns data and the covariance matrix to optimize and analyze investment portfolios.
- Adjusted weights during bearish/correction phases based on custom stock ratings.
- Portfolio allocations in the form of a pie chart.

Step 6: Monitoring FII/FPI Investment

- Converted 90-day investment data from CSV (assuming format).
- Developed a complete bar chart that clearly shows the trends and market participation rates particularly by the foreign investors.

Step 7: The Growth Opportunity Identification Process Across Industries

- Stocks were divided into independent groups depending on their respective sectors, and the average price appreciation of each group over a span of one year was calculated painstakingly.
- Displayed top 5 sectors with highest returns using a horizontal bar graph.

Step 8: Development and Integration of Web Interface

- Connected Flask with HTML templates.
- Used Chart.js to display results dynamically.
- Implemented new-generation interactivity capabilities to enable real-time data loading smoothly.

Step 9: Deployment of application over the Heroku server.

- As the final step the web application is deployed over the Heroku servers for increased accessibility.
- Below is the link.
  https://small-6cyk.onrender.com

4. CASE STUDY – CURRENT MARKET STATE

4.1 Summary

To authenticate and prove the Share Market Advisor Lite dashboard effectiveness, a case study was carried out using recent market data up to April 18, 2025. The case study establishes the system's capability to:

- Identify the current market phase with technical analysis.
- Recommend the most suitable portfolio according to the current market scenario.
- Examine foreign institutional investment (FII/FPI) trends.
- Determine leading industries for concentrated capital investment.

The dashboard integrates all these components into a single user-friendly interface, providing useful guidance based on information to retail investors.

4.2 Recent Market Data in the Dashboard

As per the real-time dashboard snapshot:

A. Market State Analysis

- Chart Shown: A line chart of Nifty 50 index values for the past 365 days.
- Observation:
    - The Nifty 50 indicates a distinct rising trend, though with occasional corrections.
    - The latest price increase puts the index above its 100-day moving average.
    - It is a sign of good momentum and bullish attitude.
    - System Interpretation: The market is "Bullish".

Implication:

- Risk appetite rises in bull markets.
- Investors have a greater likelihood of enjoying capital appreciation across industry sectors.
- The system becomes aggressive in portfolio allocation, with the aim of maximizing returns.

B. Portfolio Optimization – Markowitz Efficient Frontier

- Chart Presented: Donut chart displaying portfolio distribution.
- Expected Portfolio Return: 13.49%, a high return benchmark for the Indian equity market.

Portfolio Composition:

- Invested in 15 various companies for diversification.
- The majority of tickers (such as DRREDDY.NS, ITC.NS, BHARTIARTL.NS, HINDALCO.NS, NTPC.NS, etc.) receive 10%.

- Small allocations (1–5%) are kept for stocks such as LT. NS, ADANIPORTS.NS, and INDUSINDBK.NS, indicating cautious investment in overvalued or risky stocks.

Interpretation:

- For bearish markets, the model optimizes for maximum Sharpe Ratio, with broader sector exposure.
- Sector-wise, the low P/E ratio and healthy fundamentals stocks are chosen.
- Allocations are directed to enhance returns without enhancing risk significantly.

Risk Management:

- By limiting the maximum exposure to individual stocks, diversification minimizes portfolio-level volatility.
- Sector filters and PE ratio ensure technical allocation is accompanied by fundamental strength.

C. FII/FPI Investment Review (Last 90 Days)

- Chart Shown: Bar chart showing daily FII/FPI net investment in ₹ crores.

Observations:

- Early stage reflects net selling by foreign investors.
- Current news from April 2025 suggests further positive investment.
- There is a definite shift in the trend: from money going out to money entering.

Explanation:

- FII/FPI transactions indicate how the investors perceive the market.
- Sustained FII purchases underpin positive market recommendations and enhance liquidity.
- Positive investment flows can push equity prices higher.

Planning Smarter:

- Must align investors' exposure with FII-favored sectors (e.g., IT, Capital Goods).
- An increase in FII inflows reflects institutional faith in the medium-term trajectory of the Indian market.

D. Top 5 Industries by Growth (Last 365 Days)

- Category Displayed: Horizontal bar graph of percent growth by each sector.
- Main Sectors (approximate growth percentages):
    - Telecommunication – ~50%
    - Capital Goods – ~30%
    - Building Materials – ~28%
    - Consumer Services – ~25%
    - Financial Services – ~22%

Explanation:

- Telecommunication is the winner, suggesting favorable fundamentals or industry-wide tailwinds (e.g., 5G rollout, rural expansion).
- Capital Goods and Construction Materials reflect infra push and public/private capex boost.
- Consumer Services show rising demand and urban expenditure growth.
- Financial Services remain strong with improving credit and healthier NPAs. Investment Recommendation: At a favorable market cycle, investing in emerging industries can boost portfolio yields.
- Focusing on the top 3–5 industries, not all 15, can improve your chances for higher returns.

4.3 Summary of Observations

Feature	Observation	Inference
Market Trend	Bullish (Nifty above 100-DMA, positive momentum)	Favors risk-on strategy
Portfolio Allocation	Diversified, 13.49% return expected	Optimized for returns with fundamental support
FII/FPI Activity	Recent positive flows	External confidence, bullish confirmation
Sectoral Performance	Telecom, Capital Goods, Construction leading growth	High-conviction sectors for investment

4.4 Recommendations

According to the above facts:

1. Market Strategy:

- Leverage bullish momentum to invest aggressively but judiciously.
- choose dynamic rebalancing for shifts in the trends.

2. Portfolio Actions:

- It is recommended to take up the portfolio that has been suggested, with minor adjustments to it based on one's own risk profile and individual preferences.
- Track low-weight stocks (1%) for volatility; rebalance quarterly.

3. Area of Concentration:

- Invest in the best 3 performing sectors with higher allocations (e.g., Telecom, Capital Goods).
- Avoid over-diversifying into declining industries.

4. FII Tracking:

- Continue to track institutional action as a leading indicator.
- Treat rapid FII outflows as an indicator of caution for future turbulence.

4.5 Limitations of the Case Study

Although this specific case study provides deep insight and a substantial amount of information, it should be noted that the following limitations are relevant:

- Market condition is decided solely by price trend and moving average analysis, purely without reference to news or any important events that might be taking place.
- Portfolio optimization relies on the presumption that past returns and covariances will continue—may not apply in black swan situations.
- Sector growth is calculated on the basis of average stock returns; intra-sector volatility is not considered.
- FII reports are only valid for 90 days and don't represent long-term mood changes.

4.6 Conclusion

The case study amply illustrates the practical usefulness and analytical potential of the Share Market Advisor Lite dashboard, highlighting its effective potential in actual applications. The software gives a comprehensive view of the market's current health, enabling users to instantly comprehend the prevailing financial situation. It also presents a statistically sound investment portfolio based on robust data, enabling investors to make informed decisions. It also indicates some sectors where growth is presently focused, enabling targeted investment strategies to reap maximum returns. This groundbreaking approach, synergistically unifying the craft of visual narrative with the discipline of quantitative finance, seeks to greatly empower the retail investor. By doing this, it helps them make improved and more assertive choices in a market context sometimes challenging and harsh.

5. PERSONAL & PROFESSIONAL LEARNINGS

This project has been a highly fulfilling learning experience for me, one that has enormously expanded not only my technical expertise but also my financial understanding.

In-Depth Technical Insights

• Web development with Flask and frontend integration with Chart.js.

• Python financial data libraries (yfinance, PyPortfolioOpt).

• Implementation of quantitative finance models such as the Markowitz portfolio.

• Dynamic, real-time rendering of visualizations and charting using JavaScript.

Financial Insights and Takeaways:

• The importance of market phases interpretation, as also the use of technical indicators, plays an imperative role in formulating good decision-making.

• Risk-return trade-off balancing principles and portfolio optimization.

•Understanding of PE ratios and fundamentals of the industry.

• Determining the reason why FII investment impacts market sentiments and liquidity.

Soft Skills Learned

• Project planning under time constraints.

• Critical thinking and the ability to translate difficult information into simple, understandable graphical representations.

• Producing code that not only stays maintainable but also scalable for real-life daily usage.

• Documentation, reporting, and open communication of outcomes.

6. CONCLUSION

The "Share Market Advisor Lite" project is a long-standing and clearly defined intersection of three primary areas: financial insight, data analysis, and the field of software development. Today, with retail investors increasingly having a desire to find self-service tools that enable them, this innovative dashboard is a valuable navigation system. Its primary purpose is to assist users in making investment decisions that are well-supported by comprehensive data analysis.

Even as we need to observe that the specific model here is just a prototype that focuses on the Nifty 50 index alone, it has a framework that can easily be expanded to cover a broader range of indices. It can also be extended to cover a broader range of predictive analytics using machine learning algorithms and to include different types of sentiment analysis or pertinent macroeconomic indicators to make it more effective. This project thus does not merely fulfill its technological and analytical requirements; it also showcases the incredible and revolutionary capabilities of financial technology to empower and enable investors in their choices.

7. REFERENCES

- Johansson, C., & Nilsson, R. (2009). Visualizing real-time data designing a visual analytics tool for the stock market.
- Covaci, F., & Boscan, D. (2023). Financial Analysis Dashboard Application for Stock Exchange Listed Companies. *Journal of Computer Science and Technology Studies*, *5*(4), 10-21.
- Schwaber, K., & Beedle, M. (2002). Agile software development with Scrum. Prentice Hall.
- # Chart.js. Documentation https://www.chartjs.org/docs/latest/
- # Flask Documentation https://flask.palletsprojects.com/en/stable/

ANNEXURE 1 (Backend Code – App.py)

```
import yfinance as yf

import pandas as pd

import numpy as np

import datetime

from flask import Flask, render_template, jsonify

import csv

from pypfopt.efficient_frontier import EfficientFrontier

from pypfopt import risk_models, expected_returns

import json


app = Flask(__name__)


def get_nifty50_data():

    end_date = datetime.datetime.today()

    start_date = end_date - datetime.timedelta(days=365)

    nifty50 = yf.download("^NSEI", start=start_date, end=end_date)['Close']

    nifty50 = nifty50.dropna()

    return nifty50



def load_stock_data():

    with open('./static/nifty50list.csv', newline='') as csvfile:
```

```
        reader = csv.DictReader(csvfile)
        stocks = [row for row in reader]
    return stocks


def get_stock_data(symbols):
    end_date = datetime.datetime.today()
    start_date = end_date - datetime.timedelta(days=365)
    stock_data = {}
    for symbol in symbols:
        stock_data[symbol] = yf.download(symbol, start=start_date,
end=end_date)['Close']
    return stock_data


def get_pe_ratios(symbols):
    pe_ratios = {}
    for symbol in symbols:
        stock = yf.Ticker(symbol)
        try:
            pe_ratio = stock.info['trailingPE']
            pe_ratios[symbol] = pe_ratio
        except KeyError:
```

```
            pe_ratios[symbol] = None
    return pe_ratios



def select_min_pe_stocks(stocks, pe_ratios):
    sector_stocks = {}
    for stock in stocks:
        sector = stock['Sector']
        symbol = stock['Symbol']
        if sector not in sector_stocks:
            sector_stocks[sector] = []
        sector_stocks[sector].append((symbol, pe_ratios.get(symbol)))


    min_pe_stocks = []
    for sector, stocks_in_sector in sector_stocks.items():
        valid_stocks = [(symbol, pe) for symbol, pe in stocks_in_sector if pe is not None]
        if valid_stocks:
            min_pe_stock = min(valid_stocks, key=lambda x: x[1])
            min_pe_stocks.append(min_pe_stock)
    return min_pe_stocks


def prepare_stock_data_for_optimization(stock_data, selected_symbols):
```

```
    aligned_data = [stock_data[symbol].reindex(stock_data[selected_symbols[0]].index) for symbol in selected_symbols]



    selected_stock_data_df = pd.concat(aligned_data, axis=1)
    selected_stock_data_df.columns = selected_symbols  # Label the columns with stock symbols
    return selected_stock_data_df



def markowitz_optimization(stocks,stock_data, state):
    mu = expected_returns.mean_historical_return(stock_data)
    S = risk_models.sample_cov(stock_data)
    ef = EfficientFrontier(mu, S, weight_bounds=(0.01, 0.1))
    weights = ef.max_sharpe()
    cleaned_weights = ef.clean_weights()
    expected_return, volatility, sharpe_ratio = ef.portfolio_performance()


    adjusted_weights = cleaned_weights.copy()
    if state in["Correction State", "Bearish"]:
        ratings = {stock['Symbol']: int(stock['Ratings']) for stock in stocks}
        weighted_ratings = {symbol: cleaned_weights[symbol] * ratings.get(symbol, 3) for symbol in adjusted_weights}
```

```
        total_weighted_ratings = sum(weighted_ratings.values())
        normalized_weights = {symbol: weight / total_weighted_ratings for symbol, weight in weighted_ratings.items()}
        adjusted_weights = normalized_weights

    new_expected_return = sum(adjusted_weights[symbol] * mu[symbol] for symbol in adjusted_weights)



    chart_data = {
        "labels": list(adjusted_weights.keys()),
        "datasets": [{
            "data": [round(w * 100, 2) for w in adjusted_weights.values()]
        }]
    }

    with open("./static/chart_data.json", "w") as f:
        json.dump(chart_data, f)

    return round(new_expected_return*100,2)

def calculate_sector_growth(stocks, stock_data):
    sector_growth = {}
```

```
    for stock in stocks:
        sector = stock['Sector']
        symbol = stock['Symbol']

        if symbol in stock_data and not stock_data[symbol].empty:
            start_price = stock_data[symbol].iloc[0]
            end_price = stock_data[symbol].iloc[-1]
            growth = ((end_price - start_price) / start_price) * 100

            if sector in sector_growth:
                sector_growth[sector].append(growth)
            else:
                sector_growth[sector] = [growth]

    avg_sector_growth = {sector: np.mean(growths) for sector, growths in
sector_growth.items()}

    top_5_sectors = sorted(avg_sector_growth.items(), key=lambda x: x[1],
reverse=True)[:5]
```

```
    chart_data = {
        "labels": [sector for sector, _ in top_5_sectors],
        "datasets": [{
            "label": "Sector Growth (%)",
            "data": [round(growth, 2) for _, growth in top_5_sectors],
            "backgroundColor": ["#ff6384", "#36a2eb", "#ffce56", "#4bc0c0", "#9966ff"]
        }]
    }

    with open("./static/sector_growth.json", "w") as f:
        json.dump(chart_data, f)

@app.route('/')
def index():

    nifty_data = get_nifty50_data()
    latest_price = float(nifty_data.iloc[-1])
    highest_price = float(nifty_data.max())
    diff_percent = ((highest_price - latest_price) / highest_price) * 100.0
    mavg = float(nifty_data.rolling(window=100).mean().iloc[-1])
    print("The 100 Days Moving Average is: ",mavg)

    if latest_price > mavg:
```

```
    state = "Bullish"
    color = 'rgba(0, 255, 0, 0.5)'
elif latest_price <= mavg:
    if diff_percent <= 5.0:
        state = "Pullback State"
        color = 'rgba(255, 255, 0, 0.5)'
    elif diff_percent <= 10.0:
        state = "Correction State"
        color = 'rgba(255, 100, 0, 0.5)'
    else:
        state = "Bearish"
        color = 'rgba(255, 0, 0, 0.5)'

stocks = load_stock_data()
symbols = [stock['Symbol'] for stock in stocks]


stock_data = get_stock_data(symbols)
pe_ratios = get_pe_ratios(symbols)


min_pe_stocks = select_min_pe_stocks(stocks, pe_ratios)
selected_symbols = [stock[0] for stock in min_pe_stocks]
```

```
    selected_stock_data_df =
prepare_stock_data_for_optimization(stock_data,selected_symbols)



    expected_returns = markowitz_optimization(stocks,selected_stock_data_df, state)
    calculate_sector_growth(stocks, stock_data)


    return render_template('index.html',state=state,
color=color,alert=(state=="Bearish"),expected_return=expected_returns)


@app.route('/data')
def data():
    nifty_data = get_nifty50_data()
    data_points = {
        "labels": list(nifty_data.index.strftime('%Y-%m-%d')),
        "prices": list(map(float, nifty_data.values))
    }
    return jsonify(data_points)


if __name__ == '__main__':
    app.run()
```

ANNEXURE 2 (Frontend Code – index.html)

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="icon" href="{{ url_for('static', filename='favicon.ico') }}" type="image/x-icon">
    <title>Share Market Advisor Lite</title>
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
    <style>
        .chart-grid {
            display: flex;
            flex-wrap: wrap;
            justify-content: space-around;
        }

        .chart-container {
            position: relative;
            background-color: #ffffff;
            width: 45vw;
            height: 35vh;
```

```
            border: solid 1px green;
            border-radius: 10px;
            display: flex;
            justify-content: center;
            align-items: center;
            margin-bottom: 5px;
        }
    </style>
</head>

<body style="font-family: sans-serif; background-color: #005; color: #fff;">
    {% if alert %}
    <script>alert("Market is Bearish. Invest with Caution!");</script>
    {% endif %}
    <h2 style="text-align: center; margin-top: 15px;">Share Market Advisor Lite</h2>

    <div class="chart-grid">
        <div>
            <h4>Nifty 50 Trend (Past 365 Days) - <i style="color: #AAF;">{{ state
}}</i></h4>
            <div class="chart-container">
                <canvas id="niftyChart"></canvas>
            </div>
```

```
        </div>

        <div>
            <h4>Expected Portfolio Return: <i style="color: #AAF;">{{ expected_return
}}%</i></h4>
            <div class="chart-container">
                <canvas id="portfolioChart"></canvas>
            </div>
        </div>

        <div>
            <h4>FII/FPI Investments (Past 90 Days)</h4>
            <div class="chart-container">
                <canvas id="fiiChart"></canvas>
            </div>
        </div>

        <div>
            <h4>Top 5 Sectors by Growth (Past 365 Days)</h4>
            <div class="chart-container">
                <canvas id="sectorChart"></canvas>
            </div>
        </div>
```

```
</div>


<script>
    async function fetchData() {
        const response = await fetch('/data');
        const data = await response.json();
        return data;
    }

    async function renderChart() {
        const data = await fetchData();
        const ctx = document.getElementById('niftyChart').getContext('2d');

        new Chart(ctx, {
            type: 'line',
            data: {
                labels: data.labels,
                datasets: [{
                    label: 'Nifty 50 Closing Price',
                    data: data.prices,
                    borderColor: 'black',
                    borderWidth: 1,
                    fill: true,
```

```
            backgroundColor: '{{color}}',
            pointRadius: 0
          }]
        },
        options: {
          responsive: true,
          scales: {
            x: { title: { display: true, text: 'Date' }, grid: { display: false } },
            y: { title: { display: true, text: 'Closing Price' }, grid: { display: false } }
          },
          elements: { line: { tension: 0.2, borderWidth: 3 } }
        }
      });
}

fetch('./static/chart_data.json')
    .then(response => response.json())
    .then(data => {
        const ctx = document.getElementById('portfolioChart').getContext('2d');
        new Chart(ctx, {
            type: 'doughnut',
            data: data,
            options: {
```

```
plugins: {
    legend: {
        display: true,
        position: 'right',
        labels: {
            usePointStyle: true,
            boxWidth: 10,
            padding: 10,
            font: {
                size: 11
            },
            generateLabels: function (chart) {
                const data = chart.data;
                if (data.labels.length && data.datasets.length) {
                    return data.labels.map((label, i) => {
                        const value = data.datasets[0].data[i]+'%';
                        return {
                            text: `${label} (${value})`,
                            fillStyle: data.datasets[0].backgroundColor[i],
                            hidden: !chart.getDataVisibility(i),
                            index: i,
                            fontColor: '#777'
                        };
```

```
                });
            }
            return [];
        }
    }
},
tooltip: {
    callbacks: {
        label: function (tooltipItem) {
            let label = data.labels[tooltipItem.dataIndex] || '';
            let value = data.datasets[0].data[tooltipItem.dataIndex] || 0;
            return `${label}: ${value}%`;
        }
    }
}
},
cutout: '60%',
layout: {
    padding: {
        right: 10,
        top: 10,
        bottom: 10
    }
```

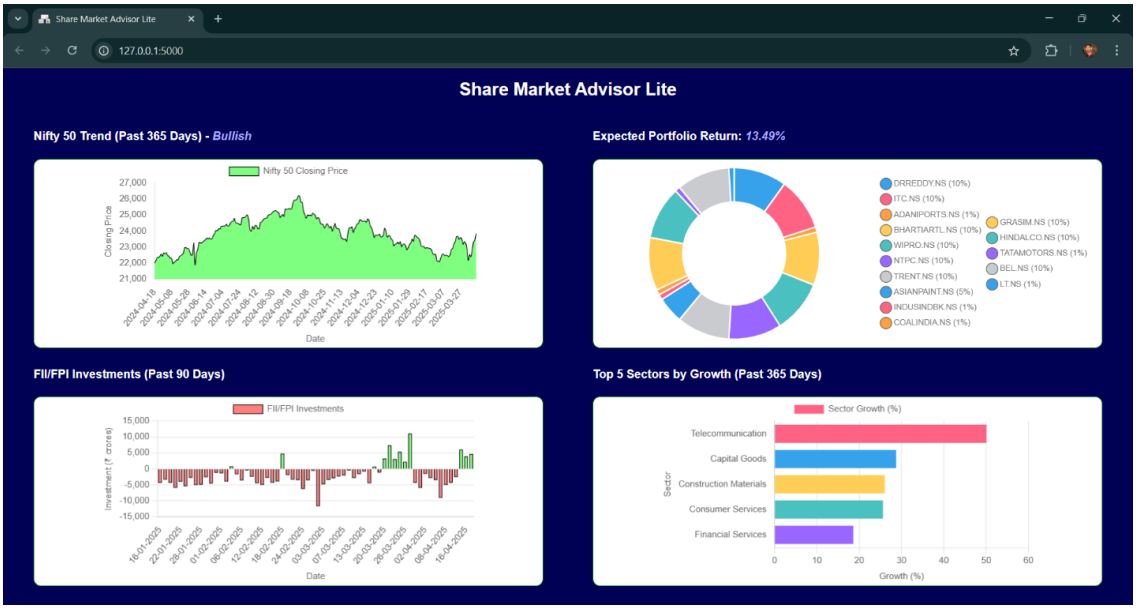
```
            },
            responsive: true,
            maintainAspectRatio: false
        }
    });
});


async function fetchFIIData() {
    const response = await fetch('/static/fii_data.csv');
    const data = await response.text();

    const rows = data.split("\n").slice(1);
    let dates = [];
    let investments = [];

    rows.forEach(row => {
        const [date, investment] = row.split(",");
        if (date && investment) {
            dates.push(date.trim());
            investments.push(parseFloat(investment.trim()));
        }
    });
```

```
        return { dates, investments };
    }

    async function renderFIIChart() {
        const data = await fetchFIIData();
        const ctx = document.getElementById('fiiChart').getContext('2d');

        new Chart(ctx, {
            type: 'bar',
            data: {
                labels: data.dates,
                datasets: [{
                    label: 'FII/FPI Investments',
                    data: data.investments,
                    backgroundColor: data.investments.map(value => value >= 0 ? 'rgba(0,
255, 0, 0.5)' : 'rgba(255, 0, 0, 0.5)'),
                    borderColor: 'black',
                    borderWidth: 1
                }]
            },
            options: {
                responsive: true,
```

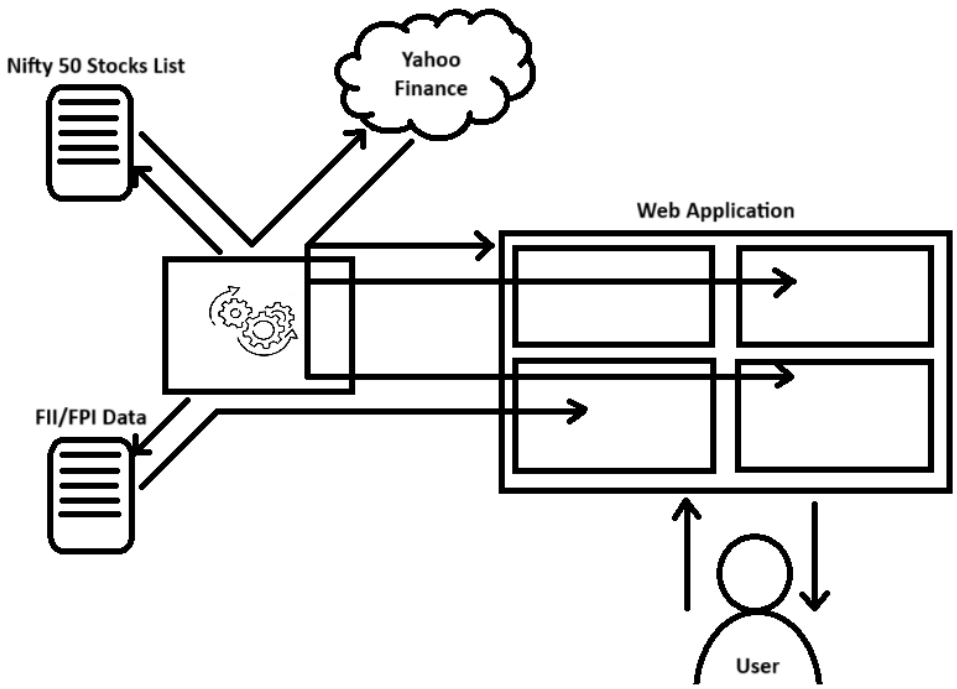
```
            scales: {
                x: { title: { display: true, text: 'Date' }, grid: { display: false } },
                y: { title: { display: true, text: 'Investment (₹ crores)' }, grid: { display:
true } }
            }
        }
    });
}

fetch('./static/sector_growth.json')
    .then(response => response.json())
    .then(data => {
        const ctx = document.getElementById('sectorChart').getContext('2d');
        new Chart(ctx, {
            type: 'bar',
            data: data,
            options: {
                indexAxis: 'y',
                responsive: true,
                scales: {
                    x: { title: { display: true, text: 'Growth (%)' }, grid: { display: true } },
                    y: { title: { display: true, text: 'Sector' }, grid: { display: false } }
                }
```

```
                }
            });
        });

        renderFIIChart();
        renderChart();
    </script>
</body>

</html>
```

ANNEXURE 3 (Project Image)



ANNEXURE 4 (Data Flow Diagram)



MRP Nikhil Kumar Vashist 23DMBA79.docx

Delhi Technological University

Document Details

Submission ID
trn:oid:::27535:96016384

Submission Date
May 15, 2025, 1:54 PM GMT+5:30

Download Date
May 15, 2025, 1:55 PM GMT+5:30

File Name
MRP Nikhil Kumar Vashist 23DMBA79.docx

File Size
289.9 KB

50 Pages

5,934 Words

37,873 Characters

5% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 10 words)

Match Groups

18 Not Cited or Quoted 5%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

5% Internet sources
1% Publications
3% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.