

# Project Report: Stock Analysis Dashboard

## 1. Introduction

The dynamic nature of financial markets requires investors and analysts to make informed decisions based on real-time and historical stock data. In this context, visual tools that help analyze trends, performance, and technical indicators are vital. This project aims to build an interactive Stock Analysis Dashboard using Python, enabling users to visualize stock prices, analyze technical trends, and export data for further research.

## 2. Abstract

This project involves the development of a web-based dashboard that performs technical analysis on stock data. Built using Streamlit, Plotly, and yfinance, the application allows users to input a stock ticker, fetch historical data, and visualize it with candlestick and line charts. Key indicators such as the Simple Moving Average (SMA), Exponential Moving Average (EMA), and Relative Strength Index (RSI) are calculated to help analyze price momentum and trends. Users can also download the historical stock data in CSV format. This tool supports individual investors, students, and analysts in better understanding market behavior through data-driven visuals.

## 3. Tools Used

Python 3 – Programming language

Streamlit – For building the interactive web app interface

yfinance – To fetch real-time and historical stock market data from Yahoo Finance

Plotly – For rendering dynamic and interactive charts (candlestick, line graphs)

Pandas – For data processing and calculation of indicators (SMA, EMA, RSI)

#### 4. Steps Involved in Building the Project

Set Up Environment: Installed required libraries (streamlit, yfinance, plotly, pandas).

Input System: Built an input box to allow users to enter stock tickers (e.g., AAPL, TSLA).

Data Fetching: Used the yfinance API to retrieve stock price data based on selected time periods and intervals.

Data Processing: Calculated:

SMA (20-day): To observe average price trends.

EMA (20-day): To respond faster to price changes.

RSI (14-day): To measure the magnitude of recent price changes.

Visualization:

Plotted candlestick charts with SMA and EMA overlays.

Displayed RSI line chart with overbought/oversold zones.

Performance Metrics: Displayed latest closing price, percentage change, and RSI.

Data Export: Enabled users to download the historical data as a .csv file.

UI/UX Improvements: Made the dashboard interactive with Streamlit widgets and error handling.

## 5. Conclusion

The Stock Analysis Dashboard effectively demonstrates how Python can be used to integrate financial data, technical analysis, and visual storytelling into a single, interactive application. This tool provides users with a powerful way to observe and analyze stock trends, make data-driven decisions, and export data for deeper analysis. With minimal dependencies and a simple UI, the dashboard serves both educational and practical purposes for those interested in the stock market.