STOCK ANALYSIS DASHBOARD

Author: Mowlya B A

Date: 26-10-2025

Abstract:

This project implements a Stock Analysis Dashboard using Python and Streamlit to fetch historical stock data and visualize technical indicators. The tool calculates Simple Moving Average (SMA), Exponential Moving Average (EMA), and Relative Strength Index (RSI) to help interpret price trends. The dashboard is interactive, enabling users to select tickers, date ranges, and indicator windows, and to export historical data — providing a compact, user-friendly analytics interface suitable for internships or demo portfolios.

Introduction:

Financial data visualization and indicator computation are core tasks in data-driven trading and analysis. Manual workflows are laborious; a lightweight dashboard improves productivity and reproducibility. This Streamlit-based dashboard aims to provide an accessible interface for fetching, computing, and visualizing stock indicators, suitable for internship demonstrations and entry-level analytics tasks.

Tools & Technologies:

Python 3.x — implementation language.

Streamlit — web UI framework for interactive apps.

yfinance — fetch historical stock data from Yahoo Finance.

pandas / numpy — data processing and numerical operations.

plotly — interactive plotting for time-series visualizations.

Steps Involved in Building the Project:

Requirements & environment: Create a virtual environment and install dependencies from requirements.txt.

UI design: A simple sidebar for inputs (ticker, start/end dates, indicator windows) and the main area for charts and tables.

Data ingestion: Use yfinance.download() to obtain OHLCV data for the requested ticker and date range.

Indicator computation: Implement SMA (rolling mean), EMA (exponential weighted mean), and RSI (standard gains/losses approach). Indicators are computed on the Close series.

Visualization: Use Plotly to render interactive price charts with indicators and a separate RSI subplot. Add a CSV download button for exporting data.

Testing: Verified with common tickers (AAPL, MSFT) and multiple ranges; handled empty or invalid ticker/date cases with user-friendly messages.

Sample Output / Features:

Interactive charts with range slider and zoom.

RSI subplot with threshold lines (70/30).

Data preview table (latest rows) and an indicator summary.

CSV export of the full fetched dataset.

Challenges & Solutions:

API reliability: yfinance depends on Yahoo Finance; to mitigate temporary failures the app shows clear error messages and lets users try another ticker or range.

Indicator edge behavior: Rolling indicators need minimal periods; code uses min_periods=1 and smooths RSI with exponential averages to avoid NaNs on small samples.

Conclusion:

The Stock Analysis Dashboard demonstrates practical skills in data ingestion, time-series indicator computation, interactive visualization, and rapid UI development using Streamlit. This project is readily extensible (add MACD, Bollinger Bands, backtesting, or deploy to Streamlit Cloud) and is a solid portfolio piece for internship interviews.