**TRENDSETTER – STOCK MARKET TREND FORECASTER DOCUMENTATION**

**1. Introduction**

**TrendSetter** is a futuristic AI-driven platform designed to help investors capture opportunities and forecast stock trends with stunning accuracy. By blending predictive modeling, time-series analysis, volatility computation, and interactive data visualization, TrendSetter empowers users to make informed investment decisions.

The system allows users to register/login, input stock ticker symbols, fetch the latest stock data, compute technical indicators like RSI and Moving Averages, analyze volatility, and apply multiple prediction models. An AI assistant provides market stance, support/resistance levels, and trading recommendations.

**2. Objectives**

* Collect real-time stock data using Yahoo Finance API (yfinance).
* Preprocess data and compute technical indicators (RSI, Moving Average) for feature enrichment.
* Analyze stock volatility and trends.
* Implement and compare multiple models:
  + Linear Regression
  + Random Forest Regressor
  + Decision Tree Regressor
  + ARIMA (Auto-Regressive Integrated Moving Average)
* Evaluate model performance using R² Score.
* Visualize RSI, moving averages, and stock price trends.
* Provide AI-powered insights including market stance, suggested entry/stoploss, and target levels.
* Deploy the system as a secure web application with interactive dashboards.

**3. Tools & Technologies**

* **Programming Language:** Python
* **Libraries:** pandas, numpy, matplotlib, seaborn, plotly, scikit-learn, statsmodels
* **Technical Analysis:** ta (Technical Analysis Library) or custom functions for RSI, MA
* **AI Models:** Gemini, GPT-4, Gamma, Mistral
* **Web Framework:** Flask or Django
* **Database:** SQLite/MySQL
* **Data Source:** Yahoo Finance (yfinance)

**4. System Architecture**

1. User Authentication – Secure Login/Register.
2. Data Collection – Fetch historical and real-time stock data.
3. Feature Engineering – Compute RSI, moving averages, and volatility metrics.
4. Preprocessing – Handle missing values, normalize data, split into training/testing sets.
5. Model Training – Train ML/ARIMA models with stock + technical indicators.
6. Prediction & Analysis – Forecast future prices.
7. AI Insights – Generate market stance, support/resistance, entry/stoploss, and target prices.
8. Visualization – Display interactive charts for price, trends, RSI, moving averages, and volatility.

**5. Features**

**5.1 Login & Register**

* Secure authentication system.
* Personalized dashboard per user.
* Role-based access control.

**5.2 Time-Series Analysis**

* Collects historical stock price data.
* Implements ARIMA, LSTM, and other time-series models.
* Predicts short-term and long-term stock trends.
* Visualizes historical vs predicted trends.

**5.3 Trend Prediction**

* Uses Linear Regression, Random Forest, and Decision Tree models.
* AI assistant provides predicted stock prices, buy/sell recommendations, and confidence levels.
* Multi-stock comparison to identify trending opportunities.

**5.4 Volatility Analysis**

* Computes mean, max, and min volatility.
* Visualizes risk and stability.
* Helps investors assess risk/reward profiles.

**5.5 AI Assistant**

* Provides personalized stock insights.
* Generates trading strategies including support/resistance levels, entry/stoploss, and target prices.
* Interactive recommendations based on real-time market data.

**6. Methodology**

**6.1 Data Collection**

* Stock data from Yahoo Finance with daily frequency.
* Features: Open, High, Low, Close, Volume, Adjusted Close.

**6.2 Relative Strength Index (RSI)**

* Momentum oscillator ranging 0–100.
* Formula: RSI = 100 - (100 / (1 + RS)), where RS = Average Gain / Average Loss (over n days).
* Interpretation: RSI > 70 → Overbought, RSI < 30 → Oversold.

**6.3 Moving Average & Volatility**

* 50-day and 200-day moving averages for trend identification.
* Volatility metrics (mean, max, min) to assess stock risk.

**6.4 Data Preprocessing**

* Convert dates to datetime format.
* Normalize numerical features.
* Split data into training (80%) and testing (20%) sets.

**6.5 Model Implementation**

* Linear Regression – Baseline model.
* Random Forest Regressor – Handles non-linearity.
* Decision Tree Regressor – Interpretable, prone to overfitting.
* ARIMA – Time-series model capturing trends.

**6.6 Model Evaluation**

* Metric: R² Score.
* Example results:
  + Linear Regression: -33.42
  + Random Forest: -2.85
  + Decision Tree: -2.84
  + ARIMA: -2.83
* RSI and MA improve feature enrichment; high market volatility still limits prediction accuracy.

**7. Web Application**

* Frontend: HTML + CSS + Chart.js / Plotly.
* Backend: Flask for data fetching, predictions, and indicator computation.
* Features:
  + Enter stock ticker.
  + Display stock price chart with RSI and MA.
  + Show volatility analysis.
  + Predict next-day price.
  + AI insights for trading strategy.
  + Automated alerts for significant changes.

**8. Results & Findings**

* RSI identifies overbought/oversold regions.
* Moving averages provide clear trend signals.
* Random Forest performed best among ML models.
* AI-powered insights provide actionable recommendations.

**9. Future Enhancements**

* Combine RSI with MACD, Bollinger Bands.
* Implement LSTM/GRU models for sequential patterns.
* Cloud deployment for real-time predictions and alerts.
* Enhance AI assistant with more sophisticated prediction models.

**10. Conclusion**

TrendSetter combines AI-driven analytics, machine learning, and technical indicators for a comprehensive stock market forecasting platform. It empowers investors with insights, predictions, and interactive visualizations, laying the foundation for future hybrid models combining AI and deep learning.

**11. References**

* Yahoo Finance API – <https://pypi.org/project/yfinance>
* Technical Analysis Library (ta) – [https://technical-analysis-library-in-python.readthedocs.io](https://technical-analysis-library-in-python.readthedocs.io/)
* Scikit-learn Documentation – [https://scikit-learn.org](https://scikit-learn.org/)
* Flask Documentation – [https://flask.palletsprojects.com](https://flask.palletsprojects.com/)