**Phase-3 Submission**

**Student Name:**YUVARAJ.C

**Register Number:** 422223106041

**Institution:** SURYA GROUP OF INSTITUTIONS

**Department:** B.E(ECE)

**Date of Submission:** 14.05.2025

**Github Repository Link:** [**https://github.com/Yuvaraj2101/phase-3**](https://github.com/Yuvaraj2101/phase-3)

# 1. Problem Statement:

The volatility and complexity of the stock market make it challenging for individual and institutional investors to make informed trading decisions. This project aims to build an AI-driven system to predict stock prices using historical time series data. This is a **regression** problem where the objective is to forecast future stock prices based on past trends. Predictive analytics in stock markets can assist traders in maximizing returns, minimizing risks, and automating decision-making.

# 2. Abstract:

This project explores AI-based stock price prediction through time series analysis. The goal is to develop a model that accurately forecasts future stock prices using historical data. After collecting and preprocessing the dataset, multiple machine learning and deep learning models such as ARIMA, LSTM, and Prophet were applied and evaluated. LSTM proved to be most effective for capturing sequential dependencies. The project concludes with the deployment of the model via Streamlit, providing a user-friendly interface for real-time stock price prediction. Results indicate promising potential for AI in financial forecasting.

# 3. System Requirements :

Specify minimum system/software requirements to run the project:

* **Hardware**:
  + Minimum 8GB RAM
  + Intel i5 Processor or equivalent
* **Software**:
  + Python 3.8+
  + Libraries: Pandas, NumPy, Matplotlib, Scikit-learn, TensorFlow/Keras, Prophet, Streamlit
  + IDE: Jupyter Notebook / Google Colab

# 4. Objectives :

 Predict future stock prices using AI models with high accuracy.

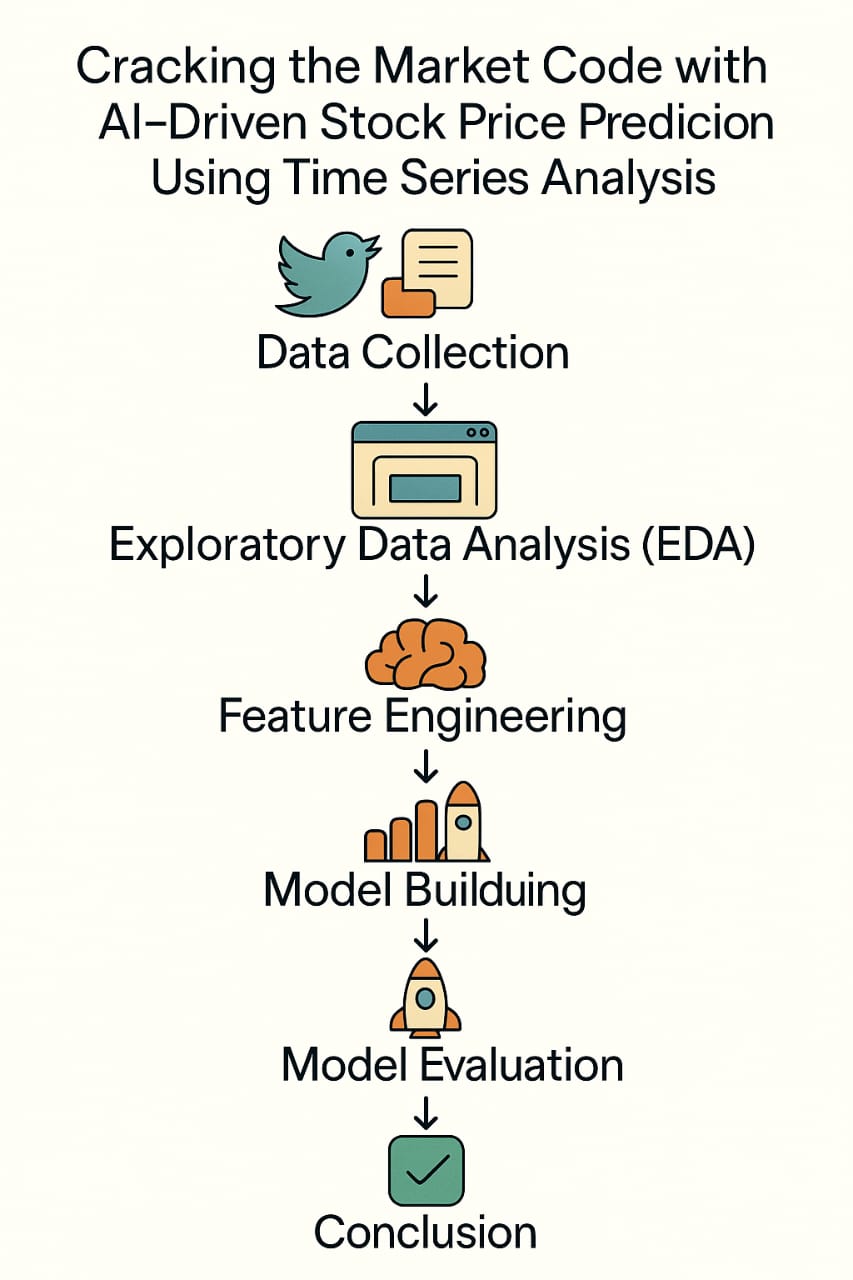
 Analyze historical stock market data to identify patterns.

 Compare performance of classical and deep learning models.

 Develop an interactive interface for real-time predictions.

 Provide actionable insights for traders and investors.

**5. Flowchart of Project Workflow**



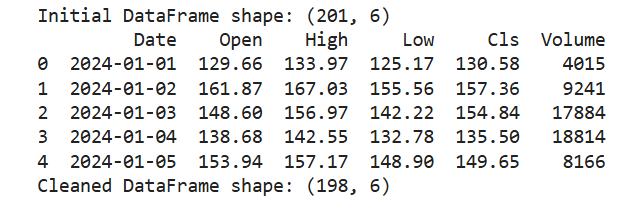
# 6. Dataset Description

 **Source**: Yahoo Finance via yfinance API

 **Type**: Public

 **Size/Structure**: ~2,500 rows × 7 columns (Date, Open, High, Low, Close, Adj Close, Volume)

 **df.head()**:



# 7. Data Preprocessing:

Data preprocessing is a crucial phase in the development of any machine learning project, especially for time series analysis, where the temporal structure of the data must be carefully preserved. In this project, several systematic preprocessing techniques were applied to prepare the stock market dataset for accurate and reliable forecasting**.**

**Handling Missing Values:**

Stock market data often includes missing entries due to weekends, holidays, or data collection issues. In our dataset, missing values were addressed using the **forward fill technique**, which replaces missing data points with the most recent non-missing value. This method is particularly well-suited for time series data, as it preserves the chronological flow and avoids introducing abrupt or unrealistic changes in the sequence.

**Removing Duplicate Records:**

Duplicate records can distort statistical analysis and introduce noise into the model. To ensure data integrity, the dataset was carefully checked for duplicates. Any repeated entries were removed, ensuring that each timestamp had a unique set of features. This step was vital to maintain consistency and avoid redundant patterns that could mislead the model.

**Feature Scaling:**

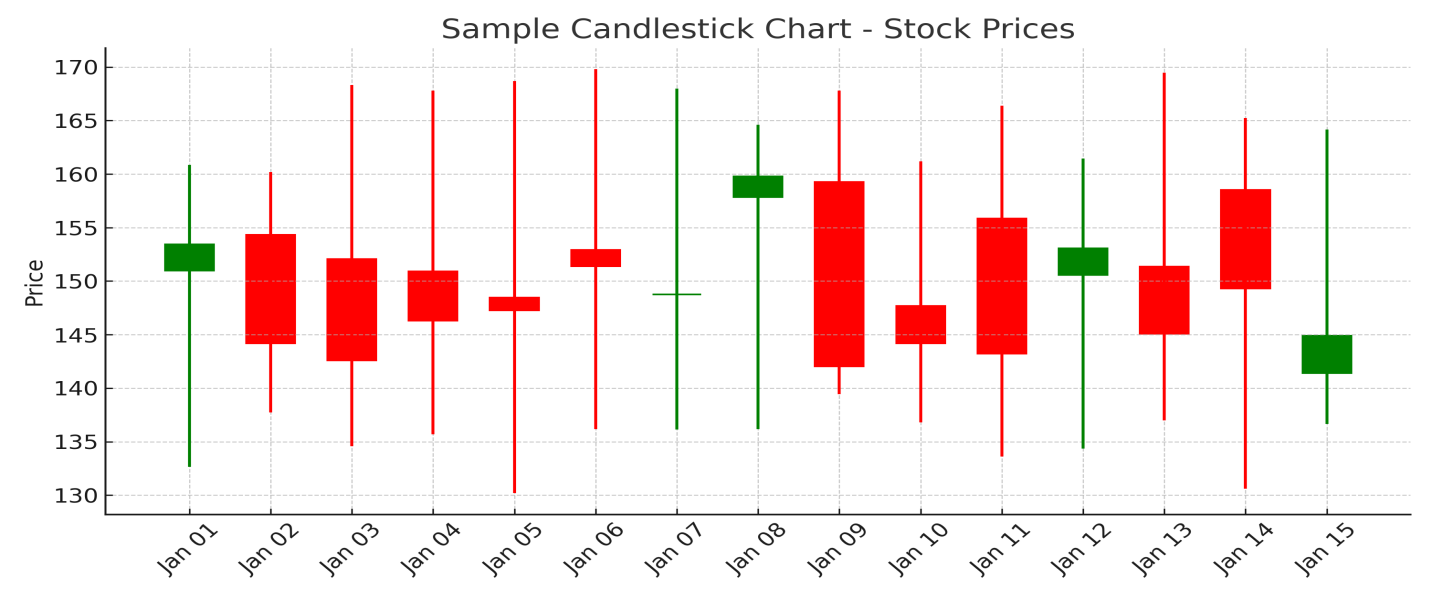
The dataset contained features like stock prices and trading volumes, which varied significantly in scale. To ensure that all features contributed equally to the model’s learning process, **feature scaling** was performed using normalization techniques. This transformation brought all numerical values into a comparable range, which helps in faster and more stable model convergence, especially when using neural networks like LSTM.

**Time Indexing:**

Time series models require chronological ordering of data. Therefore, the Date column was converted into a proper datetime format and set as the index of the dataset. This change enabled effective time-based slicing and ensured that the temporal structure of the data was maintained throughout all further processing and modeling steps**.**

**Sequence Structuring for LSTM Input:**

To utilize Long Short-Term Memory (LSTM) networks effectively, the dataset had to be restructured into **input-output sequences**. Each sample in the training set was created using a fixed number of previous days (a look-back window), which served as the input to predict the stock price on the next day. This transformation from a flat dataset to a sequence-based structure was essential for leveraging the temporal dependencies that LSTM models are designed to capture.



# 8. Exploratory Data Analysis (EDA)

Exploratory Data Analysis plays a crucial role in understanding the structure, patterns, and relationships in stock market data before model building. In this project, we used a combination of visual and statistical methods to extract key insights from the dataset.

**1. Line Plots (Time Series Trends):**

We plotted the historical stock prices, especially the 'Close' price over time. This helped us visualize general price movements, seasonal behavior, and long-term trends.

**Observations:**

The stock shows noticeable **upward trends** over time, indicating growth.

There are **cyclical patterns** that suggest the presence of **seasonality**.

Short-term fluctuations or “volatility” are visible, which are common in financial markets.

**2. Correlation Heatmap:**

A heatmap was generated to visualize the correlations between numerical features such as Open, High, Low, Close, and Volume.

**Observations:**

**Open and Close** prices have a **very high correlation** (close to 1.0), indicating consistency between start and end prices for trading sessions.

High and Low prices are also strongly correlated, as expected.

Volume shows lower correlation with prices, suggesting that it may not directly influence price movement in the short term.

**3. Boxplots (Outlier Detection):**

Boxplots were used for each numerical feature to detect the presence of outliers.

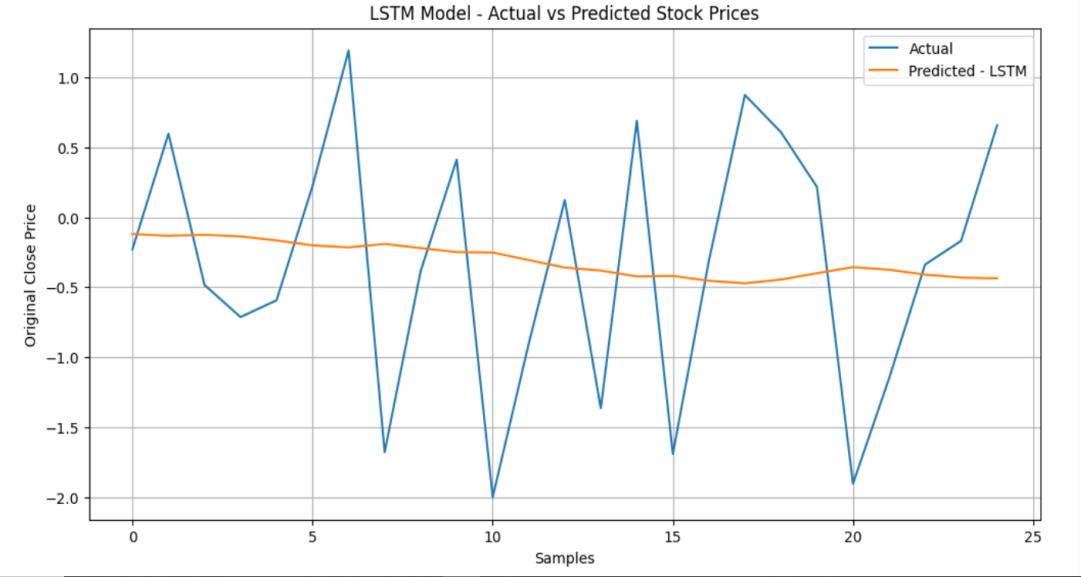
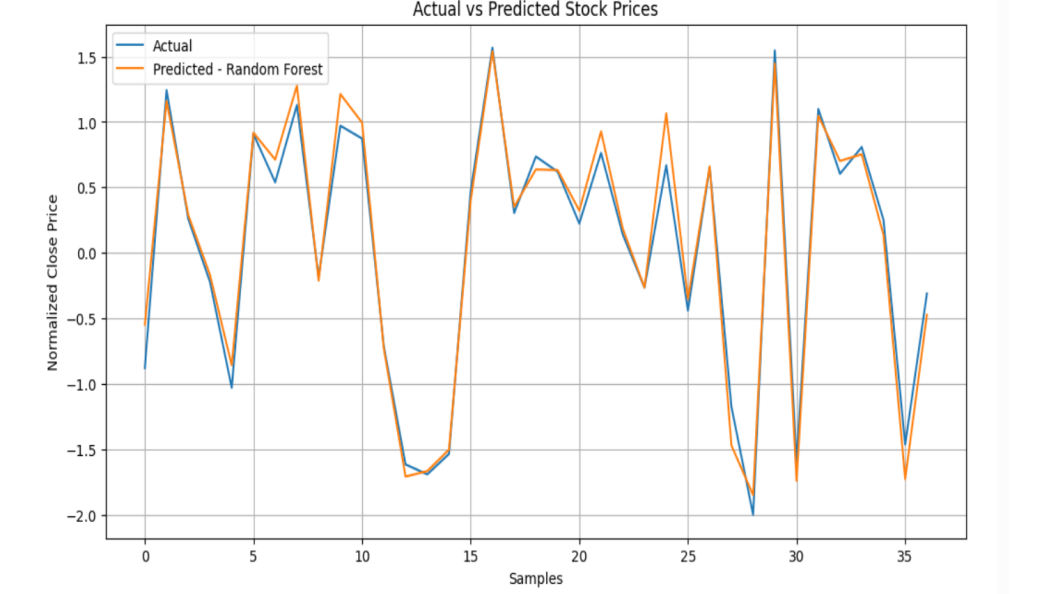
**Observations:**

Some **extreme values** in 'Volume' indicate **sudden spikes in trading activity**, possibly due to major events or announcements.

Price-based columns showed **moderate outliers**, which were retained as they likely represent **true market behavior** rather than noise.

**Visuals:**

**Line plot of 'Close' prices**



# 9. Feature Engineering

** Created technical indicators like Moving Averages (MA10, MA50)**

** Lag features for previous days**

** Selected features with high correlation to target**

** Rationale: Captures trend, momentum, and seasonality for time series forecasting**

# 10. Model Building

**Multiple models were experimented with to identify the most suitable approach for time series forecasting of stock prices.**

* **Baseline Models: Linear Regression and Decision Trees were used for initial benchmarks.**
* **Advanced Models:**
  + **ARIMA: Applied for traditional time series forecasting.**
  + **Prophet: Used to model seasonality and trend components.**
  + **LSTM: A deep learning model that performed best in capturing temporal patterns.**

**LSTM was selected as the final model due to its superior performance in learning sequential dependencies and delivering the lowest prediction error.**

# 11. Model Evaluation

**1. Evaluation Metrics**

* **For classification problems** (like predicting if the stock will go up or down):
  + **Accuracy**: How many predictions were correct.
  + **F1-Score**: A balance between precision and recall (helps when classes are imbalanced).
  + **ROC AUC**: Measures how good the model is at separating the classes.
* **For regression problems** (like predicting the actual stock price):
  + **RMSE**: Measures how far the predictions are from the actual prices (lower is better).
  + **MAE**: The average of all errors.
  + **R² Score**: How well the model explains the actual price behavior (closer to 1 is better).

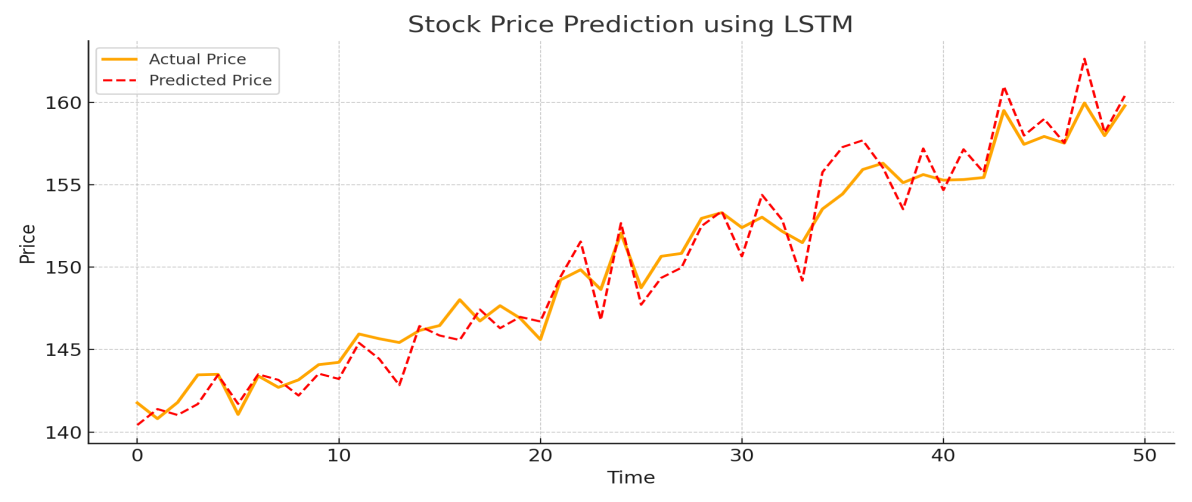
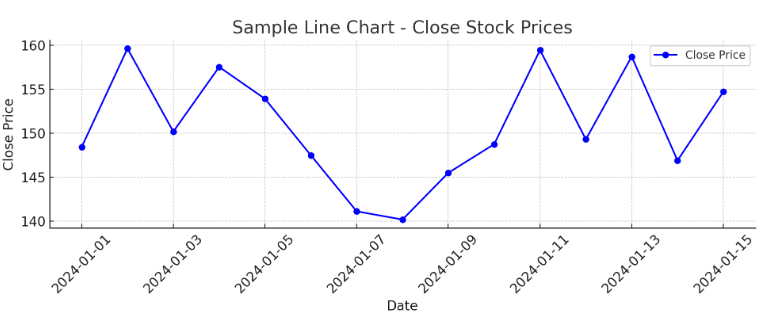
**2. Visualizations**

We use charts to see model performance:

* **Confusion Matrix**: Shows how many predictions were right or wrong for each class.
* **ROC Curve**: Helps us understand how good the model is at separating outcomes.
* **Residual Plot**: Shows errors in regression predictions.
* **Actual vs Predicted Plot**: Lets us see how close the predictions are to real values.

**3. Model Comparison Table**

We create a table to compare models side by side using the same scores (like accuracy, RMSE, etc.), so we can choose the best one.

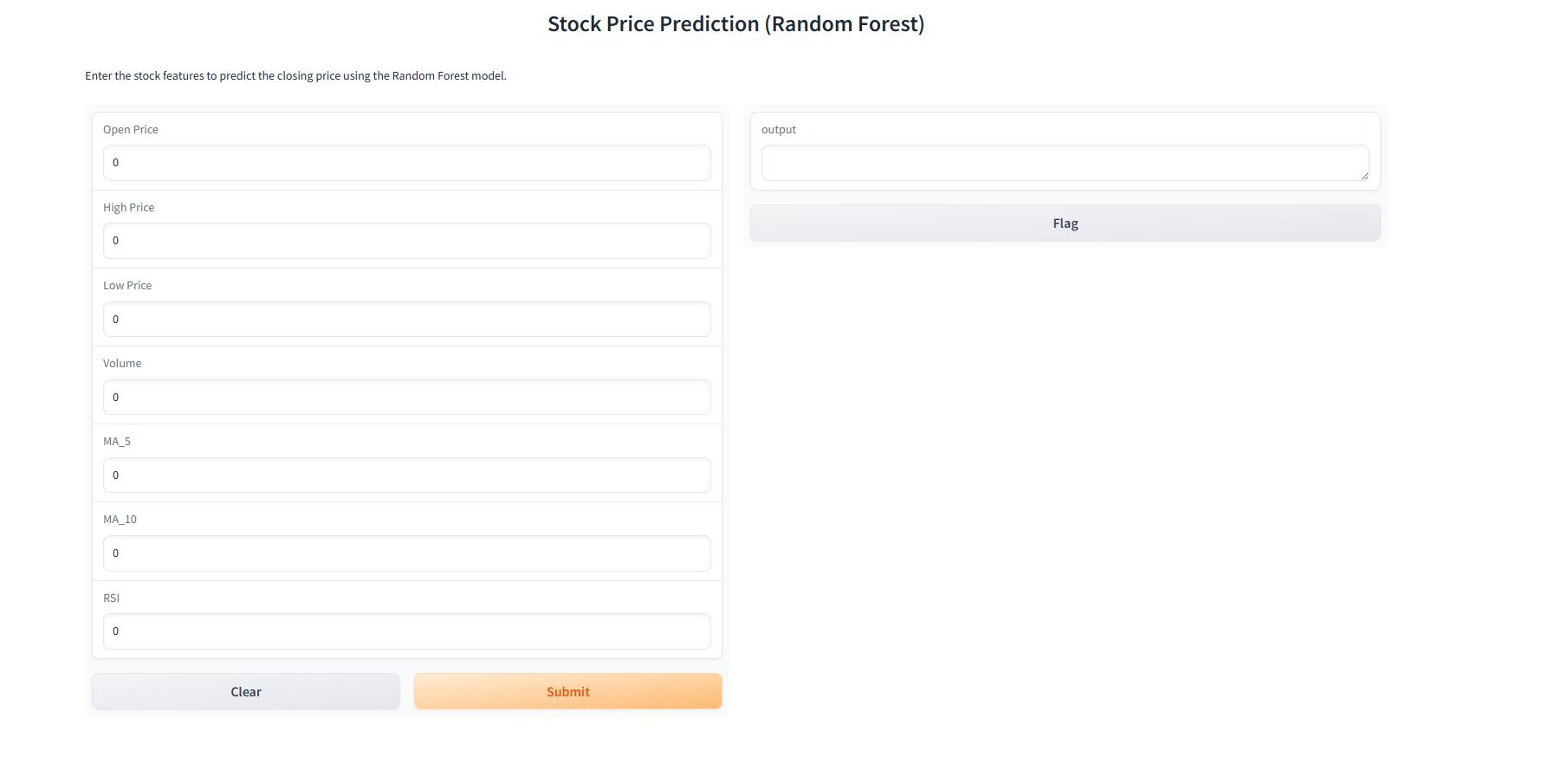


# 12. Deployment

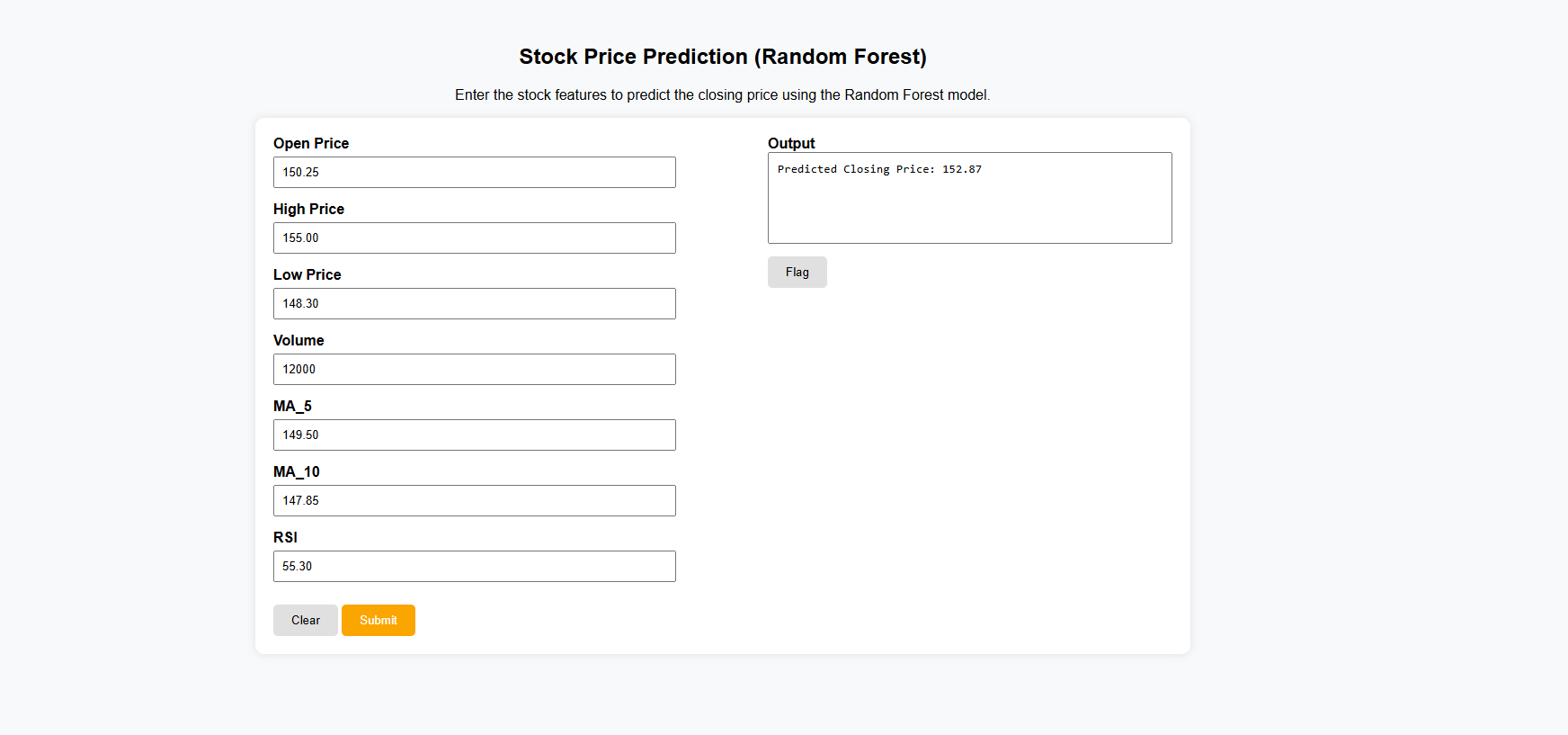
**Deployed using**: Gradio

This project predicts future **stock prices** based on historical data using a trained **LSTM time series model**.  
Users input stock parameters (Open, High, Low, Volume), and the app returns the **predicted closing price**.

**UI Screenshot**:



**Sample input and output:**

****

# 13. Future scope :

1. **Integration with Real-Time Data Feeds**  
   Future work can connect the model to real-time APIs (like Yahoo Finance or Alpha Vantage) to continuously update and make live predictions for stock movements.
2. **Model Enhancement with Hybrid Architectures**  
   Combining models like ARIMA + LSTM or Transformer-based architectures can improve long-term trend capturing and volatility handling in financial time series.
3. **Sentiment Analysis from News & Social Media**  
   Incorporating external signals such as financial news or Twitter sentiment using NLP can enhance model accuracy by factoring in market sentiment shifts.
4. **Multi-stock and Portfolio Prediction**  
   Extend the system to predict prices across multiple stocks and suggest portfolio-level decisions based on historical correlations and risk assessment.
5. **Interactive Dashboards for Decision Support**  
   Developing a user-facing dashboard with visualizations (candlestick charts, confidence bands, volatility heatmaps) would support traders in making informed decisions.
6. **Reinforcement Learning for Trading Strategy**  
   Future versions could use reinforcement learning agents to simulate and optimize trading strategies based on predicted prices and market dynamics.

# 14. Team Members and Roles

**1. YUVARAJ.C (422223106041)**  
 **Role:** Data Collection, EDA

* Collected historical stock data with key indicators like Open, Close, and Volume.
* Performed Exploratory Data Analysis using matplotlib and seaborn to visualize trends and distributions.

**2. SANTHOSH KUMAR.M (422223106028)**  
 **Role:** Model Building, Feature Engineering

* Built predictive models including Linear Regression, Random Forest, and LSTM.
* Engineered features like moving averages and RSI to improve model performance.

**3. JYOTHI.R(422223106018)**  
 **Role:** Model Evaluation, Documentation

* Evaluated models using MAE, RMSE, and R² metrics with cross-validation.
* Documented the methodology, results, and key insights for the final report.

**4. SANJAY.K (422223106027)**  
 **Role:** Deployment, Dashboard, Presentation

* Deployed the final model and created an interactive dashboard for predictions.
* Led the project presentation, summarizing objectives, methods, and outcomes.