# Microsoft Stock Forecasting using LSTM - Project Functionality

## 1. Project Overview:

Objective: The goal of this project is to forecast Microsoft stock prices based on historical stock data using an LSTM (Long Short-Term Memory) neural network model.  
  
Tools and Technologies:  
- Programming Language: Python  
- Libraries: pandas, numpy, matplotlib, scikit-learn, tensorflow/keras  
- Model Type: LSTM (Long Short-Term Memory) network, a type of Recurrent Neural Network (RNN) designed for time-series forecasting.

## 2. Data Collection and Preprocessing:

Source of Data: The stock price data for Microsoft is collected from a CSV file (`MSFT.csv`). The data includes:  
- Date: The date of the stock price record.  
- Close: The closing price of the stock.  
  
Data Preprocessing Steps:  
- Load Data: Load the dataset into a pandas DataFrame.  
- Normalize Data: Scale the data using MinMax scaling.  
- Split Data: Divide the dataset into training and testing sets (e.g., 80% training, 20% testing).  
- Sequence Preparation: Convert the data into sequential inputs, where previous days' prices predict the next day's price.

## 3. LSTM Model Architecture:

- Input Layer: Processes sequences of stock prices (e.g., the previous 60 days).  
- LSTM Layers: Extracts temporal dependencies and patterns in the sequential data.  
- Dense Layers: Fully connected layers that output predictions based on features processed by the LSTM layers.  
- Output Layer: Produces a single value, representing the predicted stock price for the next day.

## 4. Model Training and Evaluation:

- Loss Function: Mean Squared Error (MSE) is used for regression.  
- Optimizer: Adam optimizer is used to minimize the loss and adjust model weights.  
- Training Process: The model is trained over multiple epochs to optimize performance.  
- Evaluation: Compare the predicted stock prices with actual values to assess the model's accuracy.

## 5. Making Predictions and Visualizing Results:

- Predictions: Use the trained model to predict future stock prices based on test data or unseen data.  
- Visualization: Plot the actual vs. predicted prices using matplotlib to evaluate the model's accuracy visually.

## 6. Conclusion:

Insights: The LSTM model effectively captures patterns in historical stock data, enabling predictions of future prices. However, stock prices are influenced by various unpredictable factors, making predictions inherently uncertain.  
  
Future Improvements:  
- Include additional features, such as trading volume or technical indicators (e.g., moving averages).  
- Experiment with advanced models, such as LSTM with attention mechanisms.  
- Fine-tune hyperparameters, including the number of layers, neurons, and training epochs, for improved accuracy.