## **Group Assignment 1: Development Data Pre-processing Module and Web Interface**

The overall framework will consist of three main components:

1. **Data Pre-processing**: This module will be responsible for preparing and cleaning the data, ensuring it is in the proper format for trading simulations.
2. **Trading Execution**: A class that will instantiate and execute trading strategies based on the pre-processed data. This component should be able to handle different types of trading strategies and ensure accurate execution of trades.
3. **Post-Trade Analysis**: After trading simulation, this module will analyse the returns and performance of the strategy. It will calculate various metrics such as the Sharpe ratio, and generate visualisations to help in assessing the strategy’s effectiveness.

### **Objective**

The objective of this assignment is to create a robust data pre-processing class capable of analyzing stock performance data and an interactive web interface to display these analyses. This will involve both data manipulation in Python and web development.

### **Part 1: Data Pre-processing Module**

Create a Python class library with the following functionalities. Ensure the functions are modular:

1. **Fetch Data (As done in Individual assignments)**
2. **Data Characteristics**: A function that provides a summary of the dataset, including data size, number of missing values, and basic statistical measures (mean, median, standard deviation).
3. **Missing Values Handler**: Develop a function to manage missing values in the dataset. Discuss and decide within your group on the method to be used, and justify your choice.
4. **Performance Analysis**: A function that calculates the cumulative returns of a given stock and plots its performance against the Nifty index (fetch via same function here for the same period and plot against each other).

This code should be placed in a data.py file

### **Part 2: Web Interface**

Develop a web interface that asks the user for the ticker name, start date, and end date, and then:

* Displays the dataset's statistics in a tabular form.
* Indicates any missing values that were inserted.
* Plots the cumulative return graph with appropriate titles and labels.

Ensure that running main.py will serve the webpage locally and provide a user-friendly interface for inputting parameters and viewing results.

### **Final Submission Requirements**

1. **Code**: Submit a clean and well-documented main.py file that utilizes the modular code from your data.py module
2. **Web Development**: Include all necessary files to run the web interface.
3. **Demonstration Video**: Create a video (less than 5 minutes long) demonstrating the working code. The video should focus particularly on the web development aspects and explain how the interface was built and functions. Only one team member needs to present this part.