Problem Statement

Joe is starting to do Stock Trading, he has a list of companies and their closing stock prices for each day (in INR). Your task is to help him by providing a Web-Based tool where he can upload the list (CSV File) and input which company he wants to track and give date range. Based on that, you have to give at what date he should have purchased the stocks and at what date he should have sold those stocks to maximise profit/minimise loss.

Proposed Solution

1. A form screen with the following fields should be displayed
    a. Stock Name(s) - User can input 1 or more stock names to track
    b. Date range - User should select the from and to dates
    c. Input data - User should input a CSV file with closing prices for various stocks
        i. A link should be provided to download the sample CSV.

2. Validating user inputs
    a. Stock name should be valid
    b. Date range - from date should be lesser than to date
    c. File
        i. File extension should be .csv
        ii. Headers of the file should be in the given format

3. Cleaning data
    a. Convert all dates to a common format (dd-mm-yy)
    b. Check if all prices are numeric. If not discard

        [ENHANCEMENT] If the price is specified in words, convert this price to numeric values.

4. Algorithm
    a. Create a queue job with the input CSV

        In the case of huge datasets, users' time could be blocked. Hence the stock selection logic can be executed in the background. On completion, a push notification (or email if the user is registered) can be sent. On clicking the notification the results can be displayed.

        Could use Amazon SQS.

    b. For each of the selected stocks, filter records in the given date range (from and to date inclusive)

    c. Error cases

    d. Option-1: User buys and sells his/her stocks only once in the given period

        Find the 2 stocks prices x,y that have the greatest difference where
        x - stock price on buying day,
        y - stock price on selling day,

and x < y

e. Option-2: User buys and sells multiple times in the given period

Buy on the day of the 1st smallest price
Sell on day *n* when the price on $n+1^{th}$ day is less than the price on the n^{th} day and n^{th} day price is greater than buying price.
Repeat the steps for the rest of the days.
If the price after buying is lesser till the end of the range, skip that buy-sell option.

5. Displaying result

a. Display a table with the following details.

i. Stock Name
ii. Result of option 1 (single buy-sell) along with total profit for 200 shares
iii. Result of option 2 (multiple buy-sell) along with total profit for 200 shares
iv. Mean stock price
v. Standard deviation

b. Display Trend using a line graph. Plot the prices of the given stock against the given dates

c. A summary of the best stock with best buying pattern with dates to buy and sell.