STOCK VOLATILITY CALCULATOR

Submitted to the

Department of Master of Computer Applications in partial fulfilment of the requirements for the Mini Project in

Web Programming – MCA14

by

Prashanth Kumar G

1MS24MC075

Under the Guidance of

Komal S K

RAMAIAH INSTITUTE OF TECHNOLOGY

(Autonomous Institute, Affiliated to VTU)

Accredited by National Board of Accreditation & NAAC with 'A+' Grade

MSR Nagar, MSRIT Post, Bangalore-560054

www.msrit.edu

2025

RAMAIAH
Institute of Technology

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

CERTIFICATE

This is to certify that the project entitled **“Stock Volatility Calculator”** is carried out by **Prashanth Kumar G - 1MS24MC075** student of 1st semester, in partial fulfillment for the Mini Project in Web Programming - MCA14, during the academic year 2024-2025.

Komal S K	**Dr. S Ajitha**
Guide	**Head of the Department**

Name of Examiners	**Signature with Date**
1	
2	

ACKNOWLEDGEMENT

With deep sense of gratitude, I am thankful to our Respected HoD, **Dr. S Ajitha** for her pillared support and encouragement. I would like to convey my heartfelt thanks to my Project Guide **Komal S K**, Department of Master of Computer Applications, for giving me an opportunity to embark upon this topic and for her/his constant encouragement.

It gives me great pleasure to acknowledge the contribution of all people who helped me directly or indirectly realize it. First I would like to acknowledge the love and tremendous sacrifices that my Parents made to ensure that I always get an excellent education. For this and much more, I am forever in their debt.

I am thankful to all my friends for their moral and material support throughout the course of my project. Finally, I would like to thank all the Teaching and Non-Teaching Staff of Department of Master of Computer Applications for their assistance during various stages of my project work.

Prashanth Kumar G

ABSTRACT

The Stock Volatility Calculator is a web-based tool designed to help investors and financial enthusiasts analyze the volatility of stocks. The platform allows users to input stock prices for two companies over a specified timeframe (daily, weekly, or monthly) and calculates key statistical metrics such as Mean, Standard Deviation (SD), and Coefficient of Variation (CV). These metrics provide insights into the risk and stability of the stocks, enabling users to make informed investment decisions. The project is developed using HTML, CSS, and JavaScript, with additional libraries like MathJax for rendering mathematical equations and Plotly.js for interactive data visualization. The platform features a user-friendly interface, responsive design, and dynamic functionality, making it accessible across various devices.

The Stock Volatility Calculator offers a range of features to enhance stock analysis and financial decision-making. It supports Dynamic Table Generation, allowing users to input stock prices for multiple periods, and automates Statistical Calculations for Mean, Standard Deviation (SD), and Coefficient of Variation (CV) to eliminate manual errors. Interactive Charts provide a visual representation of stock trends and deviations, while Comparative Analysis enables users to assess the risk and volatility of two stocks simultaneously. The tool allows Real-Time Data Input, enabling users to modify stock prices dynamically, and features a User-Friendly Design for seamless navigation. With Performance Optimization, it ensures fast and accurate computations using efficient algorithms. Additionally, its mobile-responsive interface enhances accessibility across devices. The tool also serves as an Educational Utility, making complex financial concepts easier to understand for students and professionals. By combining automation, visualization, and ease of use, the platform makes stock volatility analysis more accessible and insightful.

The Stock Volatility Calculator demonstrates the effective use of modern web technologies to create a practical and educational tool for financial analysis. By integrating statistical methods with interactive visual elements, the project enhances the decision-making process for investors, analysts, and financial enthusiasts. The inclusion of JavaScript libraries ensures smooth user interactions and a seamless experience, making financial data analysis both accessible and insightful.

Table of Contents

1 Introduction

The Stock Volatility Calculator is a web application designed to help users analyze the volatility of stocks. Volatility is a crucial metric in finance, as it measures the degree of fluctuation in a stock's price over a given period. A highly volatile stock experiences frequent and significant price changes, making it a riskier investment, whereas a low-volatility stock is more stable and predictable. Understanding stock volatility helps investors assess risk and optimize their portfolios. This tool provides a simple, interactive, and intuitive way to calculate and visualize stock volatility, empowering users to make informed investment decisions based on statistical analysis.

The platform enables users to analyze and compare two stocks simultaneously, making it easier to evaluate relative risk and performance. Users can input stock prices over a selected timeframe, such as daily, weekly, or monthly, and the system automatically computes key statistical metrics, including Mean, Standard Deviation (SD), and Coefficient of Variation (CV)**.** These metrics provide valuable insights into stock price stability and risk exposure. By offering a comparative analysis, the tool helps users determine which stock is more stable and which carries greater uncertainty, aiding in strategic financial decision-making.

The project is developed using HTML, CSS, and JavaScript, ensuring a smooth and responsive web experience. Additional libraries such as MathJax are used for rendering mathematical equations, making financial formulas more readable and accessible. The integration of Plotly.js allows for interactive data visualization, providing dynamic charts and graphs that visually represent stock price trends and statistical results. The platform is fully responsive, adapting seamlessly to desktops, tablets, and mobile devices, ensuring a user-friendly experience across different screen sizes. With its efficient design and interactive capabilities, the Stock Volatility Calculator serves as a valuable tool for investors, students, and financial analysts seeking to understand and interpret stock market fluctuations.

1.1 Problem Definition

The Stock Volatility Calculator aims to bridge the gap between technical financial analysis and accessibility for the average investor. One of the primary challenges in stock market analysis is the lack of accessible tools that cater to non-technical users. Many existing financial platforms and tools require advanced knowledge of statistical and financial concepts, making them difficult to use for individuals without formal training. Additionally, most traditional methods involve manual calculations, which can be complex, time-consuming, and prone to errors. This makes it challenging for everyday investors to analyze stock volatility effectively.

- **Complex Calculations**

  Calculating key statistical measures such as Mean, Standard Deviation (SD), and Coefficient of Variation (CV) manually requires a solid understanding of mathematical concepts and statistical formulas. The manual computation process involves multiple steps, including data collection, squaring deviations, and applying formulas, which increases the likelihood of errors. Investors who lack mathematical expertise may struggle with these calculations, limiting their ability to assess stock risk effectively.

- **Lack of Visualization**

  Many existing financial tools rely heavily on numerical data without graphical representations, making it difficult for users to interpret trends and deviations. Without interactive visualizations, identifying patterns in stock price movements becomes challenging, especially for beginners. Visual aids such as charts and graphs are crucial in conveying complex financial information in a more digestible and intuitive manner. A lack of visual representation often leads to misinterpretation of data, making financial analysis less effective.

- **Limited Comparison**

  Most available stock analysis tools focus on analyzing a single stock at a time, without providing a comparative framework. However, for investors, comparing two stocks side by side is crucial for making informed decisions. Understanding how two different stocks perform relative to each other allows investors to assess which stock is more stable and which carries greater risk. Without a built-in comparison feature, users are forced to analyze stocks separately, making the process tedious and less efficient.

- **Non-Responsive Design**

  In today's digital age, investors and traders frequently access financial tools on mobile devices. However, many existing tools and stock analysis platforms are not optimized for smaller screens, resulting in poor user experience on smartphones and tablets. Non-responsive designs often lead to difficult navigation, improper data display, and usability issues. A lack of mobile compatibility limits accessibility, preventing users from analyzing stock volatility on the go.

2 Implementation

The Stock Volatility Calculator is implemented using a combination of HTML,CSS, and JavaScript. The development process follows a structured approach, focusing on creating a responsive and interactive user interface while ensuring accurate statistical calculations.

Development Stages

a. **Frontend Design**

- HTML is used to define the structure of the website, including input fields, tables, and result sections.

- CSS is employed to style the elements, ensuring a consistent and visually appealing design.

- JavaScript manages interactivity, such as dynamic table generation, statistical calculations, and chart rendering.

b. **Dynamic Table Generation**

- Users can input the number of periods (days, weeks, or months) and generate a table to enter stock prices for two companies.
- The table dynamically adjusts based on the selected timeframe (daily, weekly, or monthly).

c. **Statistical Calculations**

- The platform calculates key metrics such as Mean, Standard Deviation (SD), and Coefficient of Variation (CV) for both stocks.
- Mean: The average stock price over the specified period.
- Standard Deviation (SD): A measure of price volatility.
- Coefficient of Variation (CV): A normalized measure of volatility, expressed as a percentage.

d. **Interactive Charts**

- Plotly.js is used to create interactive charts that visualize stock price trends, mean lines, and deviations.
- Each chart includes annotations to highlight standard deviations from the mean.

e. **Comparative Analysis**

- The platform provides a detailed comparison of the two stocks, including their volatility and risk levels.
- Users can easily identify which stock is more volatile and make informed investment decisions.

f. **Mobile Responsiveness**

- The platform is designed to be fully responsive, ensuring a seamless user experience across devices (desktops, tablets, and smartphones).
- CSS media queries are used to adjust the layout based on screen size.

2.1 Code snippets

Stock.html

```
<html>
<head>
<title>Stock Volatility Calculator</title>
<link rel="stylesheet" href="input.css">
<link rel="stylesheet" href="calculate.css">
<link rel="stylesheet" href="graph.css">
<link rel="stylesheet" href="analysis.css">
<script type="text/javascript" async
src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.7/MathJax.js?config=TeX-MML-
AM_CHTML">
</script>
<script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
<script src="input.js"></script>
<script src="calculate.js"></script>
<script src="graph.js"></script>
<script src="analysis.js"></script>
</head>
<body>
<nav>
<h1>Stock Volatility Calculator</h1>
</nav>
<br><br>
<center><label for="company1">Enter Company 1 Name:</label>
<input type="text" id="company1" placeholder="Company 1">
<label for="company2">Enter Company 2 Name:</label>
<input type="text" id="company2" placeholder="Company 2"></center>
<center><label for="timeframe">Select Timeframe:</label>
<select id="timeframe" onchange="updateInputField()">
<option value="daily">Daily</option>
<option value="weekly">Weekly</option>
<option value="monthly">Monthly</option>
</select>
<input type="number" id="timeInput" placeholder="Enter number of periods" min="1">
<button onclick="generateTable()">Generate Table</button>
<button onclick="resetCalculator()">Reset</button></center>
<div id="tableContainer"></div>
<div id="statsContainer"></div>
<div class="equationcontainer">
```

```
<div class="equation1">
<p id="sub1"></p>
</div>
<div class="equation2">
<p id="sub2"></p>
</div>
</div>
<div class="graph-container">
<div class="graph-wrapper">
<div id="company1Chart" style="width: 820px; height: 600px;"></div>
</div>
<div class="graph-wrapper">
<div id="company2Chart" style="width: 820px; height: 600px;"></div>
</div>
</div>
<div id="analysisContainer"></div>
</body>
</html>
```

input.js

```
function generateTable() {
let period = document.getElementById("timeInput").value;
let timeframe = document.getElementById("timeframe").value;
let company1 = document.getElementById("company1").value || "Company 1";
let company2 = document.getElementById("company2").value || "Company 2";
if (period < 1) {
alert("Please enter a valid number of periods.");
return;
}
let label;
if (timeframe === "daily") {
label = "Day";
} else if (timeframe === "weekly") {
label = "Week";
} else {
label = "Month";
}
let tableHTML = `<table>
<tr>
<th>${label}</th>
<th>${company1} Price</th>
<th>${company2} Price</th>
</tr>`;
for (let i = 1; i <= period; i++) {
tableHTML += `<tr>
<td>${label} ${i}</td>
<td><input type='number' id='c1_${i}'></td>
<td><input type='number' id='c2_${i}'></td>
</tr>`;
}
tableHTML += "</table>";
document.getElementById("tableContainer").innerHTML = tableHTML;
let existingButton = document.getElementById("calcButton");
if (existingButton) {
existingButton.remove();
}
document.getElementById("tableContainer")
.insertAdjacentHTML("afterend",
"<center><button id='calcButton' onclick='calculateStats()'>Calculate
Stats</button><center>"
);

document.getElementById("statsContainer").innerHTML = "";
document.getElementById("analysisContainer").innerHTML = "";
}
```

input.css

```
* {
box-sizing: border-box;
margin: 0;
padding: 0;
}
body {
font-family: 'Segoe UI', system-ui, sans-serif;
background: #f8f9fa;
line-height: 1.6;
padding-top: 0;
margin: 20px;
background-color: #f0f4f8; /* Soft blue-gray background */
color: #2d3748;
}
nav {
background: #1b4f72;
padding: 1.5rem;
position: relative; /* Default positioning */
width: 100%;
```

```
z-index: 1000;
box-shadow: 0 2px 15px rgba(0,0,0,0.1);
transition: none; /* Remove any transitions */
}
nav h1 {
color: #fff;
font-size: 1.8rem;
text-align: center;
font-weight: 600;
}
label {
display: inline-block;
width: 180px;
margin: 15px;
font-weight: 500;
color: #495057;
}
input[type="text"],
input[type="number"],
select {
padding: 0.75rem;
border: 2px solid #e9ecef;
border-radius: 6px;
width: 250px;
margin: 10px;
font-size: 1rem;
display: inline-block;
}
button {
padding: 0.75rem 1.5rem;
margin: 15px;
background: #1b4f72;
color: white;
border: none;
border-radius: 6px;
cursor: pointer;
font-weight: 600;
transition: all 0.3s ease;
}
table {
width: 90%;
margin: 30px auto;
border-collapse: collapse;
background: white;
```

```
box-shadow: 0 1px 3px rgba(0,0,0,0.05);
}
th, td {
padding: 1rem;
text-align: center;
border: 1px solid #dee2e6;
}
th {
background: #1b4f72;
color: white;
}
@media (max-width: 768px) {
body {
padding-top: 80px;
}
label {
width: 100%;
margin: 10px 15px;
}
input[type="text"],
input[type="number"],
select {
width: calc(100% - 30px);
margin: 0 15px 10px;
}
button {
width: calc(100% - 30px);
margin: 10px 15px;
}
table {
width: 100%;
}
}
```

calculate.js

```
function calculateStats() {
let timeframe = document.getElementById('timeframe').value;
let period = document.getElementById("timeInput").value;
let company1 = document.getElementById("company1").value || "Company 1";
let company2 = document.getElementById("company2").value || "Company 2";
let c1 = [], c2 = [];
for (let i = 1; i <= period; i++) {
c1.push(parseFloat(document.getElementById(`c1_${i}`).value));
```

```
c2.push(parseFloat(document.getElementById(`c2_${i}`).value));
}
if (c1.includes(NaN) || c2.includes(NaN)) {
alert("Please enter valid numbers.");
return;
}
let sum1 = calcSum(c1), sum2 = calcSum(c2);
let mean1 = calcMean(c1), mean2 = calcMean(c2);
let sumdev1 = calcSumDev(c1, mean1), sumdev2 = calcSumDev(c2, mean2);
let sd1 = calcSD(c1, mean1), sd2 = calcSD(c2, mean2);
let cv1 = sd1 / mean1 * 100, cv2 = sd2 / mean2 * 100;
let tableComparison = `<table>
<tr><th colspan='3'>${company1}</th></tr>
<tr>
<th>Price (x<sub>i</sub>)</th>
<th>(x<sub>i</sub> - x̄)</th>
<th>(x<sub>i</sub> - x̄)<sup>2</sup></th>
</tr>`;
for (let i = 0; i < period; i++) {
let deviation1 = (c1[i] - mean1).toFixed(2);
let squared1 = (deviation1 ** 2).toFixed(2);
tableComparison += `<tr>
<td>${c1[i]}</td>
<td>${deviation1}</td>
<td>${squared1}</td>
</tr>`;
}
tableComparison += `<tr>
<td><b>∑x<sub>i</sub></b> = ${sum1.toFixed(2)}</td>
<td> </td>
<td><b>∑(x<sub>i</sub> - x̄)<sup>2</sup></b> = ${sumdev1.toFixed(2)}</td>
</tr>
</table>`;
tableComparison += `<table>
<tr><th colspan='3'>${company2}</th></tr>
<tr>
<th>Price (x<sub>i</sub>)</th>
<th>(x<sub>i</sub> - x̄)</th>
<th>(x<sub>i</sub> - x̄)<sup>2</sup></th>
</tr>`;
for (let i = 0; i < period; i++) {
let deviation2 = (c2[i] - mean2).toFixed(2);
let squared2 = (deviation2 ** 2).toFixed(2);
tableComparison += `<tr>
```

```
<td>${c2[i]}</td>
<td>${deviation2}</td>
<td>${squared2}</td>
</tr>`;
}
tableComparison += `<tr>
<td><b>∑x<sub>i</sub></b> = ${sum2.toFixed(2)}</td>
<td> </td>
<td><b>∑(x<sub>i</sub> - x̄)<sup>2</sup></b> = ${sumdev2.toFixed(2)}</td>
</tr>
</table>`;
document.getElementById("statsContainer").innerHTML = tableComparison;
document.querySelector('.equationcontainer').classList.add('visible');
document.getElementById("sub1").innerHTML = `\\[\\therefore \\text{Mean } \\bar{x} = \\
frac{\\sum x_i}{n} = \\frac{${sum1.toFixed(2)}}{${c1.length}} = ${mean1.toFixed(2)}\\]
\\[\\therefore \\text{Standard Deviation (S.D)} = \\sqrt{\\frac{\\sum \\left(x_i - \\bar{x}\\
right)^2}{n}} = \\sqrt{\\frac{${sumdev1.toFixed(2)}}{${c1.length}}} = $
{sd1.toFixed(2)}\\]
\\[\\therefore \\text{Coefficiet of Variation (C.V)} = \\frac{\\text{SD}}{x_i} = \\frac{$
{sd1.toFixed(2)}}{${mean1.toFixed(2)}} \\text{ * 100} = ${cv1.toFixed(2)} \\text{ %}\\]`;
document.getElementById("sub2").innerHTML = `\\[\\therefore \\text{Mean } \\bar{x} = \\
frac{\\sum x_i}{n} = \\frac{${sum2.toFixed(2)}}{${c2.length}} = ${mean2.toFixed(2)}\\]
\\[\\therefore \\text{Standard Deviation (S.D)} = \\sqrt{\\frac{\\sum \\left(x_i - \\bar{x}\\
right)^2}{n}} = \\sqrt{\\frac{${sumdev2.toFixed(2)}}{${c2.length}}} = $
{sd2.toFixed(2)}\\]
\\[\\therefore \\text{Coefficiet of Variation (C.V)} = \\frac{\\text{SD}}{x_i} = \\frac{$
{sd2.toFixed(2)}}{${mean2.toFixed(2)}} \\text{ * 100} = ${cv2.toFixed(2)} \\text{ %}\\]`;
MathJax.Hub.Queue(["Typeset", MathJax.Hub]);
const chartData = {
company1: {
name: company1,
prices: c1,
mean: mean1,
sd: sd1
},
company2: {
name: company2,
prices: c2,
mean: mean2,
sd: sd2
}
};
window.createCharts(chartData.company1, chartData.company2, timeframe);
analysisContainer(company1, company2, mean1, mean2, sd1, sd2, cv1, cv2);
```

```
}
function calcSum(arr) {
let sum = 0;
for (let i = 0; i < arr.length; i++) {
sum += arr[i];
}
return sum;
}
function calcMean(arr) {
let sum = 0;
for (let i = 0; i < arr.length; i++) {
sum += arr[i];
}
return sum / arr.length;
}
function calcSumDev(arr, mean) {
let sum = 0;
for (let i = 0; i < arr.length; i++) {
sum += (arr[i] - mean) ** 2;
}
return sum;
}
function calcSD(arr, mean) {
let sum = 0;
for (let i = 0; i < arr.length; i++) {
sum += (arr[i] - mean) ** 2;
}
let variance = sum / arr.length;
return Math.sqrt(variance);
}
```

calculate.css

```
.equationcontainer {
display: none;
justify-content: space-between;
width: 90%;
margin: 2rem auto;
gap: 1rem;
}
.equationcontainer.visible {
display: flex;
}
.equation1, .equation2 {
```

```
width: 45%;
display: flex;
justify-content: center;
align-items: center;
padding: 1.5rem;
background: #f7fafc;
border-radius: 8px;
border: 1px solid #e2e8f0;
margin: 0 auto;
}
#sub1, #sub2 {
width: 100%;
text-align: center;
}
.MathJax_Display {
margin: 0 auto !important;
padding: 10px 0 !important;
display: block !important;
width: fit-content !important;
}
#statsContainer {
display: flex;
gap: 2rem;
justify-content: center;
margin: 2rem 0;
flex-wrap: wrap;
}
#statsContainer table {
width: 45%;
min-width: 400px;
margin: 0;
box-shadow: 0 2px 10px rgba(0,0,0,0.1);
border-radius: 8px;
overflow: hidden;
}
@media (max-width: 768px) {
.equationcontainer {
flex-direction: column;
width: 100%;
}
.equation1, .equation2 {
width: 100%;
margin: 0.5rem 0;
}
```

```
#statsContainer table {
width: 100%;
min-width: unset;
}
}
```

graph.js

```
const config = {
displaylogo: false,
modeBarButtonsToRemove: [
'zoom2d', 'pan2d', 'select2d', 'lasso2d', 'resetScale2d',
'hoverClosestCartesian', 'hoverCompareCartesian', 'toggleSpikelines', 'resetViews'
],
modeBarButtonsToAdd: [
'toImage', 'zoomIn2d', 'zoomOut2d', 'autoScale2d'
],
displayModeBar: true
};
function createCharts(company1Data, company2Data, timeframe) {
const label = timeframe === 'daily' ? 'Day' : timeframe === 'weekly' ? 'Week' : 'Month';
const days = Array.from({length: company1Data.prices.length}, (_, i) => `${label} ${i+1}`);
Plotly.purge('company1Chart');
Plotly.purge('company2Chart');
createCompanyChart('company1Chart', company1Data, days);
createCompanyChart('company2Chart', company2Data, days);
document.querySelector('.graph-container').classList.add('active');
}
function createCompanyChart(containerId, data, days) {
const deviationLegend = {
x: [null],
y: [null],
mode: 'lines',
type: 'scatter',
name: 'Deviation',
line: { color: 'red', width: 2 },
showlegend: true
};
const priceTrace = {
x: days,
y: data.prices,
type: 'scatter',
mode: 'lines+markers',
name: 'Stock Price',
line: { color: 'blue' },
marker: { size: 8 }
};
const meanLine = {
x: days,
y: Array(days.length).fill(data.mean),
mode: 'lines',
name: 'Mean',
line: { color: 'purple', dash: 'dash' }
};
const errorLines = data.prices.map((price, i) => ({
type: 'line',
x0: days[i],
x1: days[i],
y0: data.mean,
y1: price,
line: { color: 'red', width: 2 }
}));
const annotations = data.prices.map((price, i) => ({
x: days[i],
y: (data.mean + price) / 2,
text: formatSD(price, data.mean, data.sd),
showarrow: false,
font: { color: 'black' }
}));
const layout = {
title: `<b>${data.name} Stock Price with Volatility</b>`,
xaxis: {
title: { text: `<b>${days[0].split(' ')[0]}s</b>`, font: { size: 15 } },
tickfont: { size: 13 },
showgrid: true,
gridcolor: 'rgb(175,175,175)',
linecolor: 'rgb(60,60,60)',
linewidth: 2
},
yaxis: {
title: { text: '<b>Price</b>', font: { size: 15 } },
tickfont: { size: 13 },
gridcolor: 'rgb(175,175,175)',
linecolor: 'rgb(60,60,60)',
linewidth: 2
},
shapes: errorLines,
annotations: annotations,
showlegend: true,
margin: { l: 90, r: 60, b: 80, t: 100 },
font: { family: 'Arial, sans-serif', size: 12 },
plot_bgcolor: '#fff',
paper_bgcolor: 'lightgrey'
};
Plotly.newPlot(containerId, [priceTrace, meanLine, deviationLegend], layout, config);
}
function formatSD(price, mean, sd) {
const sdValue = (price - mean) / sd;
return `${sdValue >= 0 ? '+' : ''}${sdValue.toFixed(2)} SD`;
}
window.createCharts = createCharts;
```

graph.css

```
.graph-container {
display: none;
justify-content: space-between;
align-items: flex-start;
width: 100%;
max-width: 1800px;
margin: 20px auto;
padding: 0 20px;
}
.graph-wrapper {
background: white;
padding: 1.5rem;
border-radius: 8px;
box-shadow: 0 2px 10px rgba(0,0,0,0.05);
}
.js-plotly-plot .plot-container {
border-radius: 8px;
}
.graph-container.active {
display: flex;
}
```

analysis.js

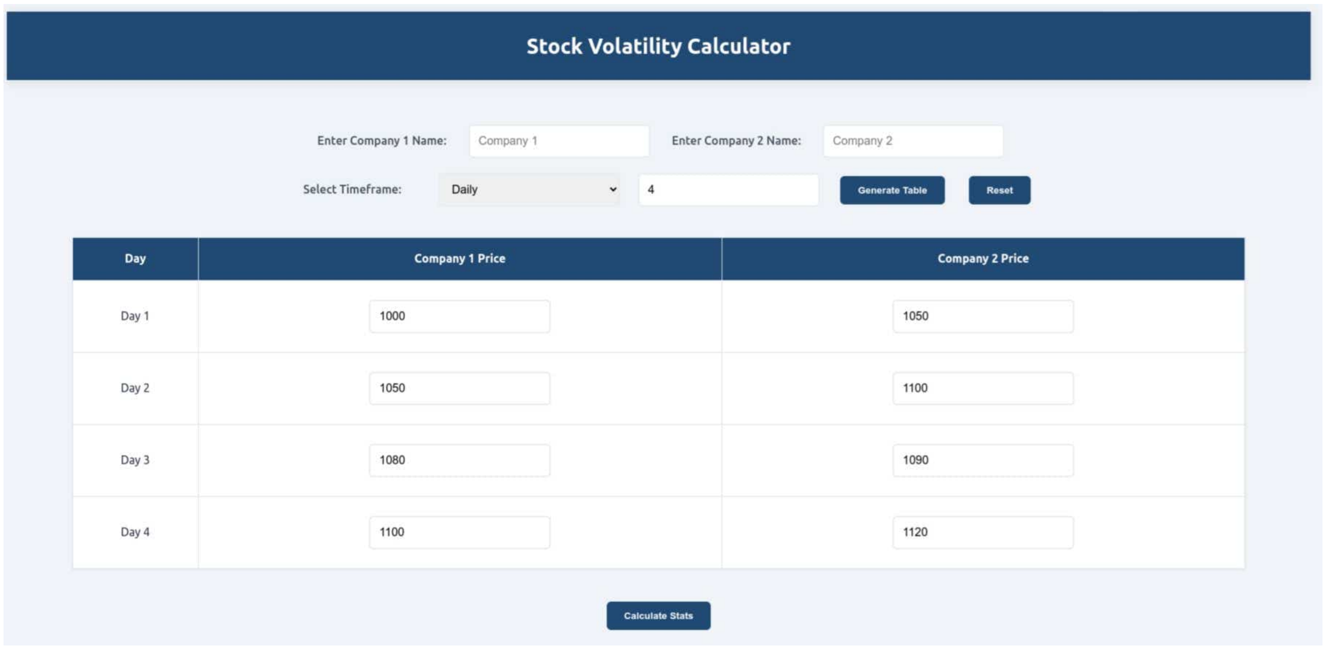
```
function analysisContainer(company1, company2, mean1, mean2, sd1, sd2, cv1, cv2) {
const isSDEqual = sd1 === sd2;
const useCV = isSDEqual;
const company1MoreVolatile = useCV ? cv1 > cv2 : sd1 > sd2;
const company2MoreVolatile = useCV ? cv2 > cv1 : sd2 > sd1;
const analysisHTML = `
<div class="report-section">
<center><h3 class="report-title">Report</h3></center>
<div class="company-comparison">
<!-- ${company1} -->
<div class="company-analysis">
<h4>${company1}</h4>
<ul class="analysis-list">
<li><strong>Mean (Average Price):</strong> ₹${mean1.toFixed(2)}</li>
<li><strong>Standard Deviation:</strong> ₹${sd1.toFixed(2)}</li>
<li><strong>Coeffiecient of Variation:</strong> ${cv1.toFixed(2)}%</li>
<li><strong>Volatility:</strong> ${isSDEqual ?
`Equal SD but ${company1MoreVolatile ? 'higher' : 'lower'} CV means $
{company1MoreVolatile ? 'more' : 'less'}` :
`${company1MoreVolatile ? 'Higher' : 'Lower'} SD means ${company1MoreVolatile ?
'more' : 'less'}`}
price swings from average mean price.</li>
<li><strong>Risk:</strong> ${company1MoreVolatile ? 'More' : 'Less'} price swings means
${company1MoreVolatile ? 'less' : 'more'} stability and ${company1MoreVolatile ? 'higher' :
'lower'} risk.</li>
<li><strong>Returns:</strong> ${company1MoreVolatile ?
'High volatility means higher potential returns with higher risk' :
'Low volatility means stable returns with lower risk'}.</li>
<li class="conclusion-item"><strong>Conclusion:</strong> ${company1MoreVolatile ?
`If you are looking for higher potential returns and are willing to take on more risk on
investment, then ${company1} is the better choice` :
`If you are looking for stable returns with consistent growth and less risk on investment, then
${company1} is the better choice`}.</li>
</ul>
</div>
<!-- ${company2} -->
<div class="company-analysis">
<h4>${company2}</h4>
<ul class="analysis-list">
<li><strong>Mean (Average Price):</strong> ₹${mean2.toFixed(2)}</li>
<li><strong>Standard Deviation:</strong> ₹${sd2.toFixed(2)}</li>
<li><strong>Coeffiecient of Variation:</strong> ${cv2.toFixed(2)}%</li>
<li><strong>Volatility:</strong> ${isSDEqual ?
`Equal SD but ${company2MoreVolatile ? 'higher' : 'lower'} CV means $
{company2MoreVolatile ? 'more' : 'less'}` :
`${company2MoreVolatile ? 'Higher' : 'Lower'} SD means ${company2MoreVolatile ?
'more' : 'less'}`}
price swings from average mean price.</li>
```

```
<li><strong>Risk:</strong> ${company2MoreVolatile ? 'More' : 'Less'} price swings means
${company2MoreVolatile ? 'less' : 'more'} stability and ${company2MoreVolatile ? 'higher' :
'lower'} risk.</li>
<li><strong>Returns:</strong> ${company2MoreVolatile ?
'High volatility means higher potential returns with higher risk' :
'Low volatility means stable returns with lower risk'}.</li>
<li class="conclusion-item"><strong>Conclusion:</strong> ${company2MoreVolatile ?
`If you are looking for higher potential returns and are willing to take on more risk on
investment, then ${company2} is the better choice` :
`If you are looking for stable returns with consistent growth and less risk on investment, then
${company2} is the better choice`}.</li>
</ul>
</div>
</div>
</div>
`;
document.getElementById("analysisContainer").innerHTML = analysisHTML;
}
function resetCalculator() {
document.getElementById("company1").value = "";
document.getElementById("company2").value = "";
document.getElementById("timeInput").value = "";
document.getElementById("tableContainer").innerHTML = "";
let existingButton = document.getElementById("calcButton");
if (existingButton) {
existingButton.remove();
}
document.querySelector('.equationcontainer').classList.remove('visible');
document.querySelector('.graph-container').classList.remove('active');
document.getElementById("statsContainer").innerHTML = "";
document.getElementById("sub1").innerHTML = "";
document.getElementById("sub2").innerHTML = "";
Plotly.purge('company1Chart');
Plotly.purge('company2Chart');
}
```

analysis.css

```
.report-section {
background: white;
border-radius: 8px;
box-shadow: 0 2px 15px rgba(0,0,0,0.05);
padding: 2rem;
margin: 2rem auto;
max-width: 1200px;
}
.company-comparison {
display: grid;
grid-template-columns: repeat(auto-fit, minmax(400px, 1fr));
gap: 2rem;
margin-top: 1.5rem;
}
.company-analysis {
background: #fff;
padding: 1.5rem;
border-radius: 8px;
box-shadow: 0 2px 10px rgba(0,0,0,0.05);
border-left: 4px solid #1b4f72;
}
.company-analysis h4 {
color: #1b4f72;
margin: 0 0 1rem 0;
font-size: 1.25rem;
padding-bottom: 0.5rem;
border-bottom: 2px solid #e9ecef;
}
.analysis-list li {
margin-bottom: 0.75rem;
padding: 0.5rem 0;
border-bottom: 1px solid #f1f3f5;
}
.conclusion-item {
background: #f8f9fa;
padding: 1rem;
border-radius: 6px;
margin-top: 1rem;
color: #1b4f72;
font-weight: 600;
}
```

3 User Interface Screenshots



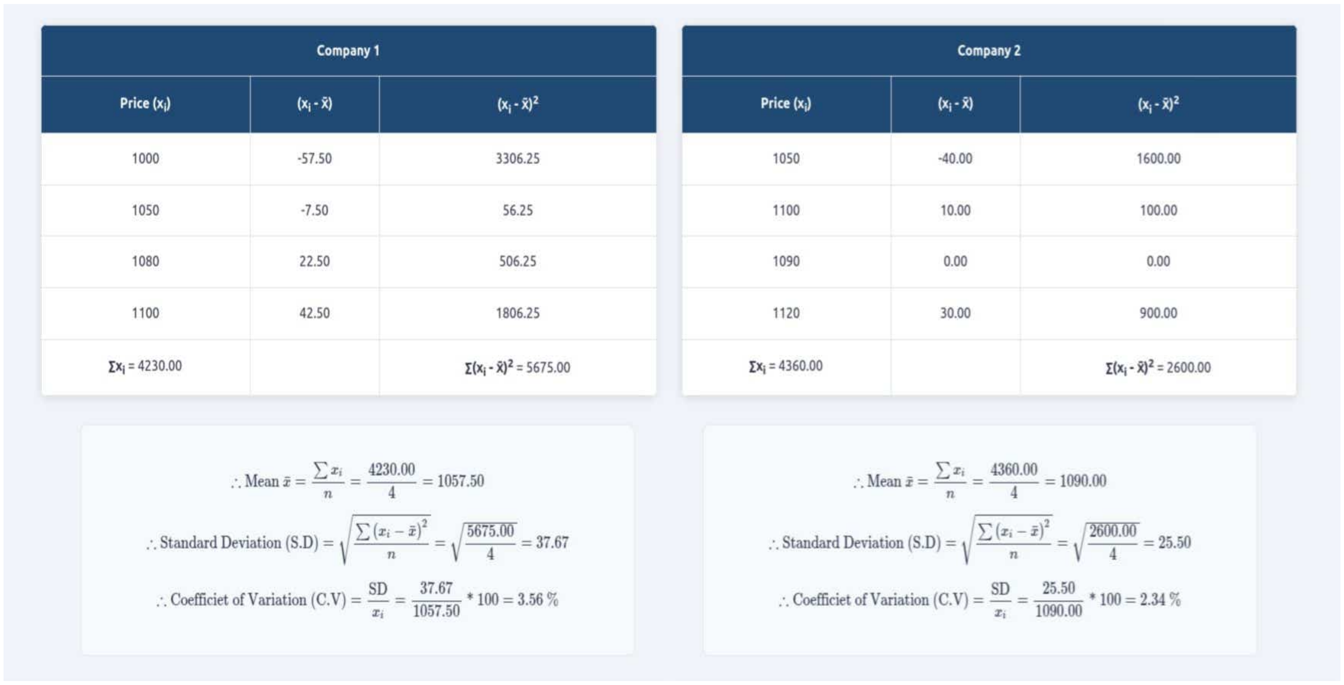
Stock Volatility Calculator

Enter Company 1 Name: Company 1 Enter Company 2 Name: Company 2

Select Timeframe: Daily 4

Day	Company 1 Price	Company 2 Price
Day 1	1000	1050
Day 2	1050	1100
Day 3	1080	1090
Day 4	1100	1120

Figure 1: Input Section



Company 1		
Price (x_i)	(x_i - $\bar{x}$)	(x_i - $\bar{x}$)^2
1000	-57.50	3306.25
1050	-7.50	56.25
1080	22.50	506.25
1100	42.50	1806.25
$\Sigma x_i = 4230.00$		$\Sigma(x_i - \bar{x})^2 = 5675.00$

Company 2		
Price (x_i)	(x_i - $\bar{x}$)	(x_i - $\bar{x}$)^2
1050	-40.00	1600.00
1100	10.00	100.00
1090	0.00	0.00
1120	30.00	900.00
$\Sigma x_i = 4360.00$		$\Sigma(x_i - \bar{x})^2 = 2600.00$

$$\therefore \text{Mean } \bar{x} = \frac{\sum x_i}{n} = \frac{4230.00}{4} = 1057.50$$

$$\therefore \text{Standard Deviation (S.D)} = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n}} = \sqrt{\frac{5675.00}{4}} = 37.67$$

$$\therefore \text{Coefficiet of Variation (C.V)} = \frac{\text{SD}}{x_i} = \frac{37.67}{1057.50} * 100 = 3.56\ \%$$

$$\therefore \text{Mean } \bar{x} = \frac{\sum x_i}{n} = \frac{4360.00}{4} = 1090.00$$

$$\therefore \text{Standard Deviation (S.D)} = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n}} = \sqrt{\frac{2600.00}{4}} = 25.50$$

$$\therefore \text{Coefficiet of Variation (C.V)} = \frac{\text{SD}}{x_i} = \frac{25.50}{1090.00} * 100 = 2.34\ \%$$

Figure 2: Statistical Analysis

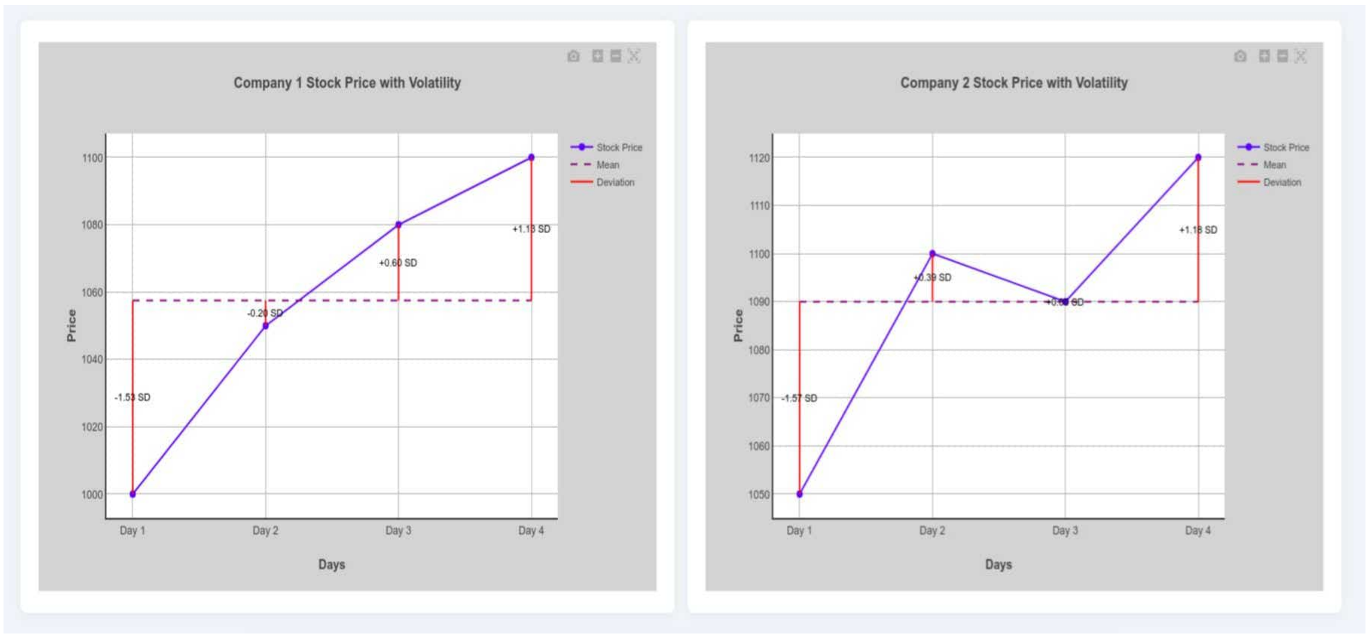
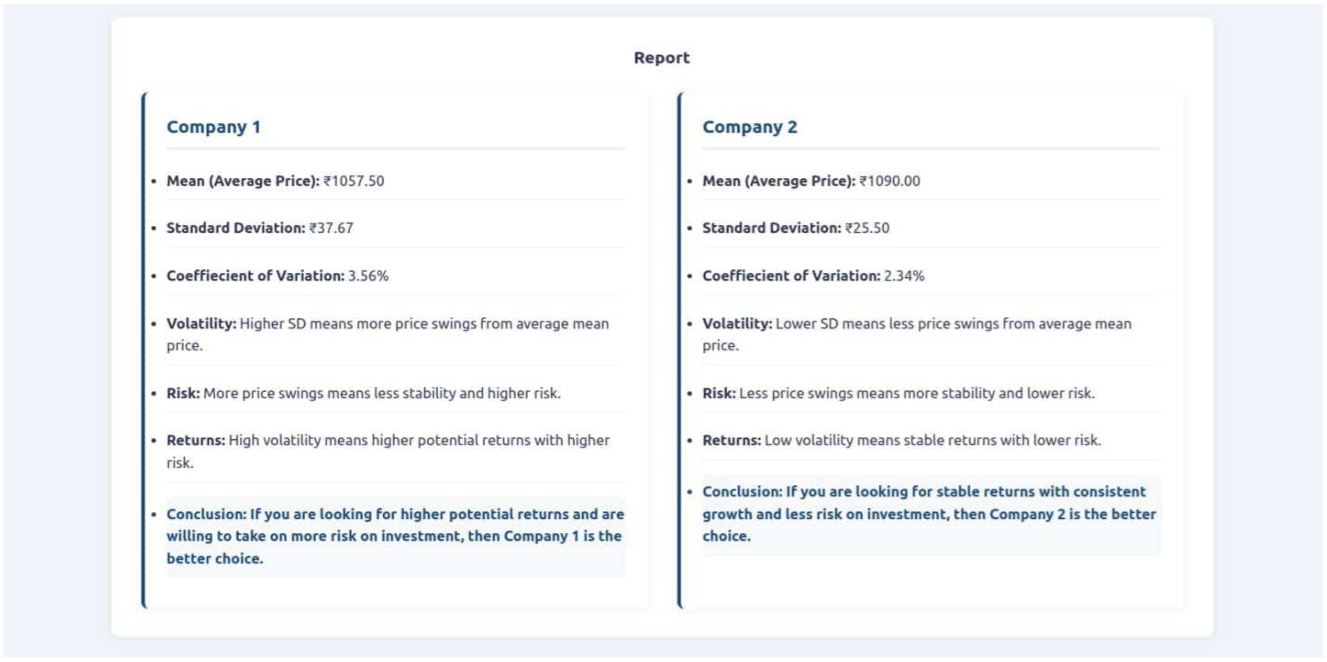


Figure 3: Interactive Graphs



Report

Company 1

- **Mean (Average Price):** ₹1057.50
- **Standard Deviation:** ₹37.67
- **Coeffiecient of Variation:** 3.56%
- **Volatility:** Higher SD means more price swings from average mean price.
- **Risk:** More price swings means less stability and higher risk.
- **Returns:** High volatility means higher potential returns with higher risk.
- **Conclusion: If you are looking for higher potential returns and are willing to take on more risk on investment, then Company 1 is the better choice.**

Company 2

- **Mean (Average Price):** ₹1090.00
- **Standard Deviation:** ₹25.50
- **Coeffiecient of Variation:** 2.34%
- **Volatility:** Lower SD means less price swings from average mean price.
- **Risk:** Less price swings means more stability and lower risk.
- **Returns:** Low volatility means stable returns with lower risk.
- **Conclusion: If you are looking for stable returns with consistent growth and less risk on investment, then Company 2 is the better choice.**

Figure 4: Comparative Analysis

4 Conclusion and Future Enhancement

Conclusion

The Stock Volatility Calculator stands as a robust and intuitive tool designed to simplify financial analysis for users of all expertise levels. By leveraging modern web technologies like HTML, CSS, and JavaScript, the platform transforms complex statistical calculations into an accessible and interactive experience. The core functionality revolves around automating the computation of critical metrics such as Mean, Standard Deviation, and Coefficient of Variation, which are essential for assessing stock volatility. The integration of Plotly.js enables dynamic visualizations of price trends and deviations, while MathJax ensures that mathematical formulas are rendered clearly, enhancing transparency in the calculation process.

The responsive design guarantees seamless usability across devices, from desktops to smartphones, ensuring that users can analyze data anytime, anywhere. Beyond its technical achievements, the project serves an educational purpose, demystifying financial concepts for novice investors and providing a practical tool for seasoned analysts. By bridging the gap between theoretical finance and real-world application, the Stock Volatility Calculator empowers users to make informed, data-driven decisions, fostering confidence in navigating the complexities of stock markets. This project exemplifies how thoughtful design and technical precision can democratize access to sophisticated financial tools, making volatility analysis approachable and actionable for a broader audience.

Future Enhancement

To further enhance the utility and scope of the Stock Volatility Calculator, the following improvements are proposed:

- **Real-Time Data Integration**

  Integrate APIs like Alpha Vantage or Yahoo Finance to fetch live stock prices, reducing manual data entry and enabling real-time analysis.

- **Advanced Financial Metrics**

  Expand calculations to include metrics like Beta (market risk), Sharpe Ratio (risk-adjusted returns), and Monte Carlo Simulations for predictive analysis.

- **Portfolio Analysis**

  Allow users to compare volatility across multiple stocks (beyond two) and create custom portfolios to assess overall risk.

- **Exportable Reports**

  Add options to export results as PDF or Excel files, enabling users to save or share their analyses.

- **Machine Learning Predictions**

  Use ML models to predict future volatility trends based on historical data, enhancing decision-making capabilities.