**What Is This Project?**

This is a **simple financial risk analysis tool**. You pick a stock (like Apple), and it:

1. Gets recent stock price data.
2. Calculates:
   * **Daily returns** (how much it goes up or down each day)
   * **Volatility** (how risky/unstable it is)
   * **Value at Risk (VaR)** (what’s the worst-case daily loss with 95% confidence)
3. Shows a **graph** of those daily returns.

**📁 File 1: main.py**

This is the **main program** — it brings everything together and runs it.

python

CopyEdit

from risk\_metrics import get\_data, calculate\_returns, calculate\_volatility, calculate\_var

from visualizer import plot\_returns

* This **imports** the functions we wrote in other files.
* get\_data: Downloads stock prices
* calculate\_returns: Figures out daily % changes
* calculate\_volatility: Measures how wild the stock moves
* calculate\_var: Estimates how much money you might lose on a bad day
* plot\_returns: Draws a line graph of those % changes

python

CopyEdit

def main():

ticker = 'AAPL'

* 'AAPL' is Apple’s stock symbol. You could change this to 'MSFT' for Microsoft, etc.

python

CopyEdit

data = get\_data(ticker)

* This downloads the past 1 year of prices for Apple.

python

CopyEdit

returns = calculate\_returns(data)

* This converts those prices into **percentage changes per day**.

python

CopyEdit

vol = calculate\_volatility(returns)

var = calculate\_var(returns)

* These two lines **calculate the risk**:
  + vol → how much prices bounce around (risk)
  + var → how bad a loss you might expect on a worst day (VaR)

python

CopyEdit

print(f"Annualized Volatility for {ticker}: {vol:.2%}")

print(f"95% Value at Risk (1-day): {var:.2%}")

* These **print the results** in percentages.

python

CopyEdit

plot\_returns(returns, ticker)

* This shows a **nice graph** of the returns using Plotly.

python

CopyEdit

if \_\_name\_\_ == "\_\_main\_\_":

main()

* This tells Python: “If you run this file, run the main() function.”

**📁 File 2: risk\_metrics.py**

This file **does the calculations** and data downloading.

**1. Download stock prices:**

python

CopyEdit

import yfinance as yf

import numpy as np

* yfinance: a library that pulls stock data from Yahoo Finance
* numpy: helps with math (like standard deviation)

python

CopyEdit

def get\_data(ticker='AAPL', period='1y', interval='1d'):

data = yf.download(ticker, period=period, interval=interval, progress=False)

return data['Adj Close']

* This function gets 1 year (1y) of daily prices (1d) for the stock.
* Adj Close is the adjusted price for splits/dividends — more accurate.
* Returns a list of those prices.

**2. Calculate returns:**

python

CopyEdit

def calculate\_returns(price\_series):

return price\_series.pct\_change().dropna()

* .pct\_change() computes daily percentage changes.
* .dropna() removes the first day (which will be NaN because there's no previous day to compare).

**3. Volatility:**

python

CopyEdit

def calculate\_volatility(returns):

return np.std(returns) \* np.sqrt(252)

* This calculates **standard deviation** (volatility).
* Multiplies by √252 to get annualized risk (since there are ~252 trading days/year).

**4. Value at Risk (VaR):**

python

CopyEdit

def calculate\_var(returns, confidence\_level=0.95):

return np.percentile(returns, (1 - confidence\_level) \* 100)

* Gets the **5th percentile** (bottom 5%) of the return values.
* So if VaR = -3%, there’s a 5% chance of losing 3% or more in one day.

**📁 File 3: visualizer.py**

This creates a chart using Plotly.

python

CopyEdit

import plotly.graph\_objs as go

def plot\_returns(returns, ticker):

fig = go.Figure()

fig.add\_trace(go.Scatter(y=returns, name=f'{ticker} Daily Returns'))

* go.Scatter: makes a line chart
* y=returns: plots the daily return values

python

CopyEdit

fig.update\_layout(title=f'{ticker} Daily Returns',

xaxis\_title='Date',

yaxis\_title='Return',

template='plotly\_dark')

* Adds titles to the chart.
* Makes it dark-themed using plotly\_dark.

python

CopyEdit

fig.show()

* Opens the chart in your browser!

**📁 File 4: requirements.txt**

This lists the Python libraries your project needs.

nginx

CopyEdit

yfinance

pandas

numpy

plotly

When someone runs:

bash

CopyEdit

pip install -r requirements.txt

It installs everything listed.

**✅ Summary**

You now have:

* A real-time dashboard for analyzing a stock’s daily returns and risks.
* Code that:
  + Pulls stock data
  + Calculates financial risk metrics
  + Visualizes results interactively