

Finzome Assessment – Yashika Tirkey

Contact -9987198692/tirki.yashika@gmail.com

Libraries used-

- Pandas
- NumPy
- os
- flask

Web Framework-

- Flask

Main.py

```
import pandas as pd
from flask import Flask, request, render_template, jsonify

from calculations import calculations_daily_volatility,
calculations_annualized_volatility

app = Flask(__name__)

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/upload', methods=['POST'])
def upload_calculation():
    try:
        # Check if CSV file is uploaded
        print('Entered into calculations part')
        file = request.files['file']
        if file and file.filename.endswith('.csv'):
            # Process the CSV file using Pandas
            df = pd.read_csv(file, encoding='unicode_escape')
            # Calculate daily volatility
            daily_volatility = calculations_daily_volatility(df)
            # Calculate annualized volatility
            annualized_volatility =
calculations_annualized_volatility(daily_volatility, df)
            # Return the results in JSON format
            return jsonify({"daily_volatility": daily_volatility,
                            "annualized_volatility":
annualized_volatility})

        except Exception as e:
            return jsonify({"error": str(e)}), 500

if __name__ == '__main__':
    app.run(debug=True)
```

calculations.py

```
import numpy as np
import pandas as pd

def calculations_daily_volatility(data: pd.DataFrame):
    data['Daily Returns '] = data['Close '].pct_change()
    Daily_Volatility = np.std(data['Daily Returns '].dropna())
    return Daily_Volatility

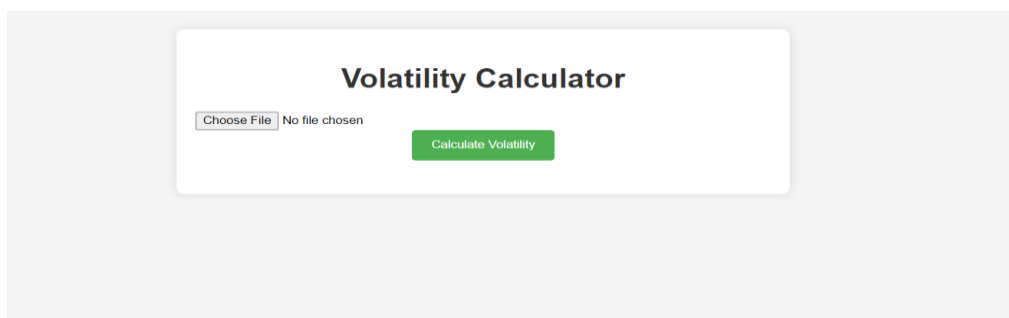
def calculations_annualized_volatility(daily_volatility: float, data:
pd.DataFrame):
    length_of_data = len(data)
    Annualized_Volatility = daily_volatility * length_of_data
    return Annualized_Volatility
```

the Flask app exposes an endpoint `/upload` that accepts a CSV file, calculated daily volatility and annualized volatility are returned in JSON format.

The **Calculations.py** file contains Python code responsible for calculating volatility. Specifically, it implements formulas for daily returns, daily volatility, and annualized volatility using a dataset. The calculations are performed based on the closing prices provided in the dataset.

The **Main.py** file utilizes the functionality implemented in **Calculations.py** to process a dataset and present the results in JSON format. This Python script imports the volatility calculations from **Calculations.py** and incorporates them into a Flask application. The application exposes an endpoint that accepts either a CSV file or a file path as parameters. Upon receiving the data, it computes the daily and annualized volatility, presenting the results in a structured JSON format.

Step 1



The screenshot shows a web interface titled "Volatility Calculator". It features a file upload section with a "Choose File" button and the text "No file chosen". Below this is a green "Calculate Volatility" button.

Step 2: Upload CSV file

Volatility Calculator

Choose File

NIFTY 50.csv

Calculate Volatility

Step 3: final Output

Volatility Calculator

Choose File

NIFTY 50.csv

Calculate Volatility

Daily Volatility: 0.0062
Annualized Volatility: 1.5490