QUERY 1-

5.1 For your chosen stock, calculate the mean daily return and daily standard deviation of returns, and then just annualise them to get mean expected annual return and volatility of that single stock. (annual mean = daily mean * 252 , annual stdev = daily stdev * sqrt(252))

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```
data = pd.read_csv('BAJFINANCE.csv')
data = data[data.Series=='EQ']
data.dropna(inplace=True)
data=data[['Close Price']]
```

In [3]:
```
data
```

Out[3]:

	Close Price
0	1332.95
1	1347.75
2	1324.80
3	1314.55
4	1289.15
5	1242.15
6	1233.75
7	1224.35
8	1258.85
9	1317.80
10	1307.45
11	1321.55
12	1326.95
13	1327.35
14	1361.25
15	1363.10
16	1336.65
17	1343.15
18	1361.70
19	1376.85
20	1364.95
21	1371.50
22	1361.00
23	1405.25
24	1410.30
25	1399.85
26	1409.60
27	1416.65
28	1401.80
29	1400.75
...	...
465	2945.25
466	2995.85
467	3025.00

	Close Price
468	3001.45
469	3055.20
470	3046.00
471	3039.45
472	3114.20
473	3024.10
474	3008.70
475	2998.35
476	3047.85
477	3008.80
478	3022.25
479	3032.50
480	3014.45
481	2991.25
482	3035.05
483	3093.05
484	3081.00
485	3096.85
486	3095.95
487	3131.75
488	3111.85
489	3034.30
490	3017.05
491	2921.30
492	2971.35
493	2922.85
494	2931.85

494 rows × 1 columns

In [4]:
```
returns = data.pct_change()
mean_return = returns.mean()
return_stdev = returns.std()
annualised_return = round(mean_return * 252,2)
annualised_stdev = round(return_stdev * np.sqrt(252),2)
print('The annualised mean return of the stock is',annualised_return)
print ('The annualised volatility is', annualised_stdev)
```

```
The annualised mean return of the stock is Close Price    0.45
dtype: float64
The annualised volatility is Close Price    0.32
dtype: float64
```

QUERY 2-

5.2 Now, we need to diversify our portfolio. Build your own portfolio by choosing any 5 stocks, preferably of different sectors and different caps. Assume that all 5 have the same weightage, i.e. 20% . Now calculate the annual returns and volatility of the entire portfolio (Hint : Don't forget to use the covariance)

In [7]:
```
d1= pd.read_csv('BAJFINANCE.csv')
d2= pd.read_csv('SUZLON.csv')
d3= pd.read_csv('RELIANCE.csv')
d4= pd.read_csv('BPCL.csv')
d1=d1[d1.Series=='EQ']
d2=d2[d2.Series=='EQ']
d3=d3[d3.Series=='EQ']
d4=d4[d4.Series=='EQ']
d1.dropna(inplace=True)
d2.dropna(inplace=True)
d3.dropna(inplace=True)
d4.dropna(inplace=True)
data=pd.concat([d1['Close Price'], d2['Close Price'], d3['Close Price'], d4['Clos
data.columns=['BAJFINANCE','SUZLON','RELIANCE','BPCL']
```

In [8]:
```
data
```

Out[8]:

	BAJFINANCE	SUZLON	RELIANCE	BPCL
0	1332.95	19.60	1344.10	727.85
1	1347.75	19.70	1356.30	733.30
2	1324.80	19.90	1353.10	727.45
3	1314.55	20.00	1327.35	714.90
4	1289.15	20.60	1318.85	702.45
5	1242.15	20.40	1324.15	695.60
6	1233.75	19.75	1305.55	699.55
7	1224.35	18.85	1297.70	711.90
8	1258.85	19.35	1302.45	726.85
9	1317.80	19.80	1337.30	749.65
10	1307.45	19.15	1356.90	754.85
11	1321.55	19.15	1359.35	738.75
12	1326.95	19.30	1340.70	742.65
13	1327.35	19.50	1328.25	739.00
14	1361.25	19.75	1324.75	724.80
15	1363.10	19.85	1329.35	728.55
16	1336.65	19.35	1312.80	725.55
17	1343.15	19.35	1339.00	717.30
18	1361.70	19.15	1331.70	706.85
19	1376.85	19.20	1335.70	703.75
20	1364.95	19.00	1319.45	701.70
21	1371.50	18.95	1314.35	699.00
22	1361.00	18.75	1357.50	692.00
23	1405.25	18.95	1383.95	670.40
24	1410.30	18.75	1388.80	673.80
25	1399.85	18.55	1409.25	667.15
26	1409.60	18.60	1412.35	664.75
27	1416.65	18.70	1422.00	654.40
28	1401.80	18.70	1432.65	641.80
29	1400.75	18.40	1435.85	630.75
...	...	...	...	...
465	2945.25	6.50	1349.25	378.95
466	2995.85	6.15	1360.00	385.85
467	3025.00	6.20	1363.25	397.55

	BAJFINANCE	SUZLON	RELIANCE	BPCL
468	3001.45	6.80	1391.85	391.90
469	3055.20	6.50	1389.70	380.50
470	3046.00	6.70	1375.20	363.20
471	3039.45	6.70	1353.05	354.30
472	3114.20	6.65	1353.90	358.10
473	3024.10	6.70	1329.25	351.00
474	3008.70	6.70	1334.45	356.30
475	2998.35	6.70	1331.40	357.75
476	3047.85	7.20	1346.80	361.70
477	3008.80	7.35	1343.10	361.15
478	3022.25	7.35	1340.15	358.40
479	3032.50	7.10	1343.75	357.50
480	3014.45	6.80	1385.95	362.90
481	2991.25	7.20	1345.35	340.10
482	3035.05	7.25	1363.85	336.35
483	3093.05	7.15	1389.50	349.00
484	3081.00	7.20	1372.40	358.25
485	3096.85	6.85	1392.80	371.05
486	3095.95	6.75	1392.80	379.85
487	3131.75	6.75	1405.05	378.45
488	3111.85	6.60	1408.85	380.75
489	3034.30	6.35	1384.90	390.35
490	3017.05	5.95	1343.50	378.85
491	2921.30	5.65	1299.45	379.80
492	2971.35	6.40	1256.45	368.05
493	2922.85	5.60	1251.15	362.95
494	2931.85	NaN	1232.05	355.10

495 rows × 4 columns

In [10]:
```
returns = data.pct_change()

mean_daily_returns = returns.mean()
cov_matrix = returns.cov()

weights = np.asarray([0.25,0.25,0.25,0.25])

portfolio_return = round(np.sum(mean_daily_returns * weights) * 252,2)

portfolio_std_dev = round(np.sqrt(np.dot(weights.T,np.dot(cov_matrix, weights))))

print('The annualised mean return of the stock is ',portfolio_return)
print('The annualised volatility is ', portfolio_std_dev)
```

```
The annualised mean return of the stock is  -0.05
The annualised volatility is  0.27
```

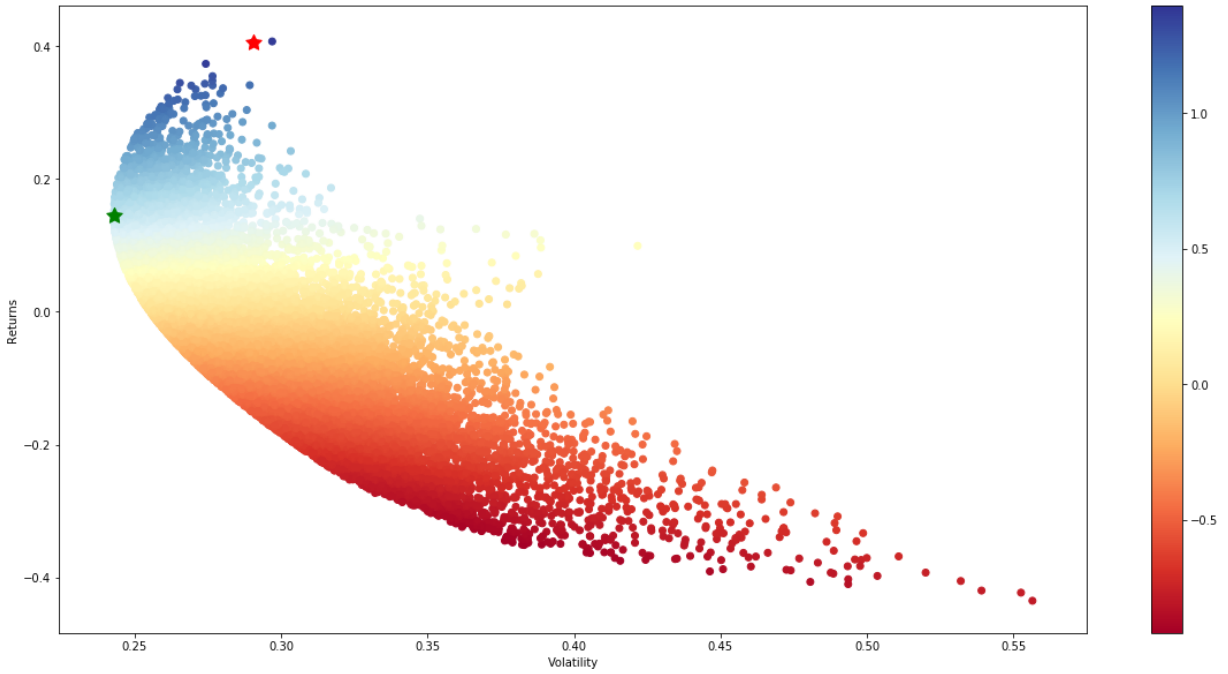
QUERY 3-

5.3 Prepare a scatter plot for differing weights of the individual stocks in the portfolio , the axes being the returns and volatility. Colour the data points based on the Sharpe Ratio (Returns/Volatility) of that particular portfolio. QUERY 4-

5.4 Mark the 2 portfolios where - Portfolio 1 - The Sharpe ratio is the highest Portfolio 2 - The volatility is the lowest.

In [11]:
```
from matplotlib.pyplot import figure
figure(figsize=(20,10))
num_portfolios= 25000
results = np.zeros((3+data.columns.size, num_portfolios))
for i in range(num_portfolios):
    weights = np.array(np.random.random(data.columns.size))
    weights/= np.sum(weights)
    portfolio_return = np.sum(mean_daily_returns * weights)*252
    portfolio_std_dev = np.sqrt(np.dot(weights.T, np.dot(cov_matrix, weights)))*
    results[0,i]=portfolio_return
    results[1,i]=portfolio_std_dev
    results[2,i]=results[0,i]/results[1,i]
    for j in range(len(weights)):
        results[j+3,i]=weights[j]
results_frame =pd.DataFrame(results.T,columns=['ret','stdev','sharpe','BAJFINANC
max_sharpe_port = results_frame.iloc[results_frame['sharpe'].idxmax()]
min_vol_port = results_frame.iloc[results_frame['stdev'].idxmin()]
plt.scatter(results_frame.stdev, results_frame.ret,c=results_frame.sharpe,cmap='
plt.xlabel('Volatility')
plt.ylabel('Returns')
plt.colorbar()
plt.scatter(max_sharpe_port[1],max_sharpe_port[0],marker=(5,1,0),color='r',s=200
plt.scatter(min_vol_port[1],min_vol_port[0],marker=(5,1,0),color='g',s=200)
```

Out[11]: <matplotlib.collections.PathCollection at 0x1c5da5b10b8>



In [ ]: