In [2]:
```
#import the libraries
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestRegressor
import matplotlib.pyplot as plt
```

In [3]:
```
df = pd.read_csv("C:/Users/HP/Desktop/RL_DAILY_DATASET (1).csv")
df = df.dropna()

#Look at the data
df
```

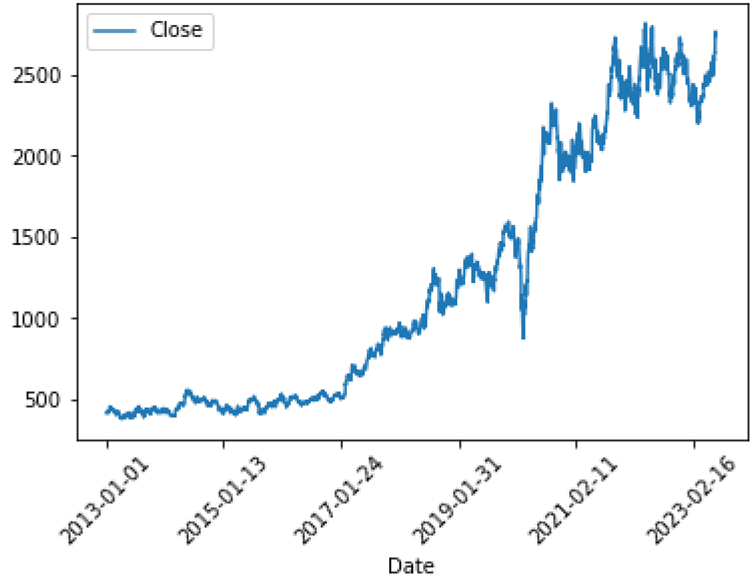
Out[3]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2013-01-01	418.037415	419.325226	415.610443	416.402924	387.885101	3152667.0
1	2013-01-02	418.037415	423.981079	417.319244	419.993866	391.230164	6203434.0
2	2013-01-03	420.315826	426.952911	418.334595	426.333771	397.135895	7968629.0
3	2013-01-04	426.903381	428.240692	422.767578	426.878632	397.643463	6140890.0
4	2013-01-07	428.785553	431.410645	421.900787	424.278259	395.221130	7064261.0
...	...	...	...	...	...	...	...
2597	2023-07-10	2688.899902	2756.000000	2675.000000	2735.050049	2735.050049	15340262.0
2598	2023-07-11	2752.899902	2770.000000	2737.600098	2764.699951	2764.699951	9262001.0
2599	2023-07-12	2766.300049	2802.000000	2761.649902	2767.750000	2767.750000	8645662.0
2600	2023-07-13	2783.899902	2799.000000	2737.250000	2743.000000	2743.000000	6776172.0
2601	2023-07-14	2750.000000	2760.899902	2725.100098	2740.699951	2740.699951	6979790.0

2600 rows × 7 columns

In [4]:
```
#show the data visually
df.plot(x="Date", y = "Close")
plt.xticks(rotation = 45)
```

Out[4]:
```
(array([-500.,    0.,  500., 1000., 1500., 2000., 2500., 3000.]),
 [Text(-500.0, 0, '2021-07-09'),
  Text(0.0, 0, '2013-01-01'),
  Text(500.0, 0, '2015-01-13'),
  Text(1000.0, 0, '2017-01-24'),
  Text(1500.0, 0, '2019-01-31'),
  Text(2000.0, 0, '2021-02-11'),
  Text(2500.0, 0, '2023-02-16'),
  Text(3000.0, 0, '')])
```



In [5]:
```python
#create the model
model = RandomForestRegressor()
```

In [6]:
```python
#Train the model
X = df[['Open', 'High', 'Low', 'Volume']]
X = X[:int(len(df)-1)]
y = df['Close']
y = y[:int(len(df)-1)]
model.fit(X,y)
```

Out[6]:
```
RandomForestRegressor()
```

In [7]:
```python
predictions = model.predict(X)
print('Model score is: ', model.score(X,y))
```

```
Model score is:  0.9999719308976864
```

In [8]:
```
#Test the model
predictions = model.predict(X)
print('The model score is: ', model.score(X,y))
```

```
The model score is:  0.9999719308976864
```

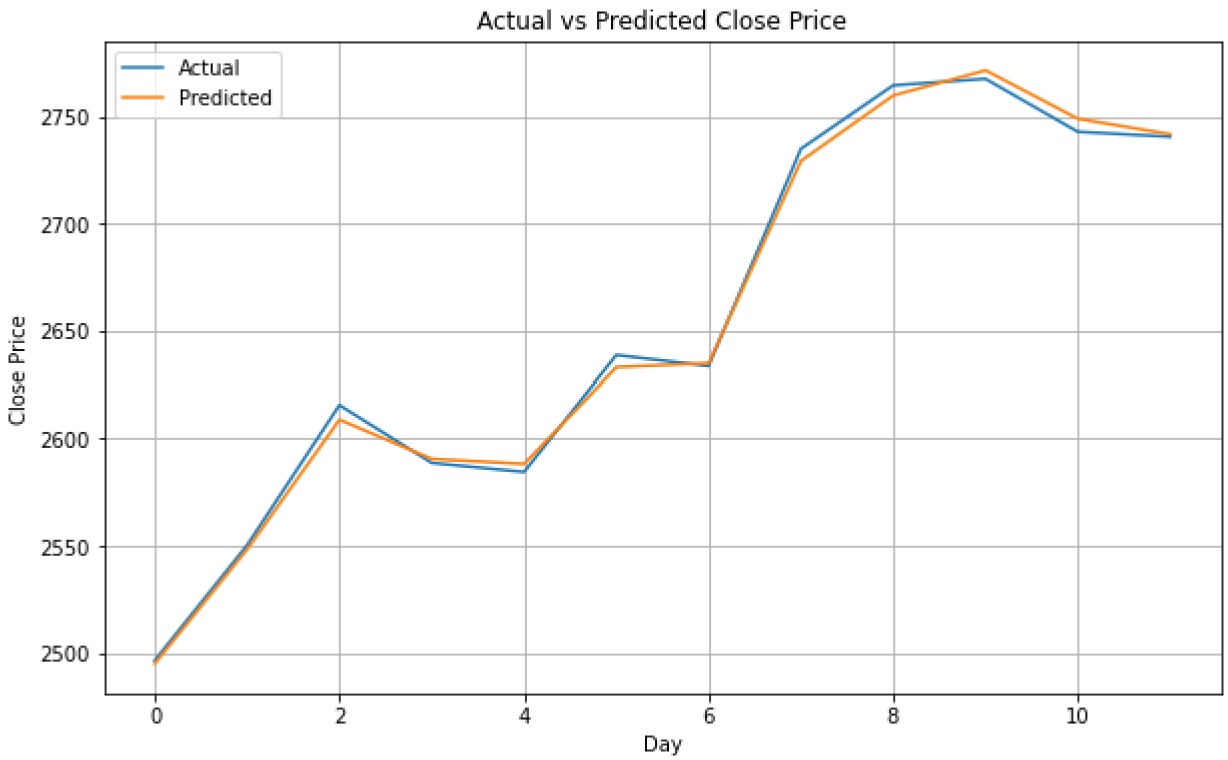
In [10]:
```
# Get the first 12 rows of data for prediction
last_12_days_data = df[['Open', 'High', 'Low', 'Volume']].tail(12)

# Make predictions for the first 12 days
predictions = model.predict(last_12_days_data)

print('The model predicts the last 12 rows (or days) to be: ', predictions)

# Print actual values for comparison
actual_values = df['Close'].tail(12).values
print('Actual values for the last 12 days:')
print(actual_values)
```

```
The model predicts the last 12 rows (or days) to be:  [2494.12649657 2549.6525147  2608.92797354 2591.26951911 2587.01802989
 2631.28349363 2635.42156027 2727.34553479 2759.52348873 2770.7750049
 2751.44550784 2741.21303969]
Actual values for the last 12 days:
[2496.449951 2550.25     2615.699951 2588.75     2584.5      2638.75
 2633.600098 2735.050049 2764.699951 2767.75     2743.       2740.699951]
```

In [26]:
```
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error, r2_score

# Assuming 'predictions' and 'actual_values' are available from the previous code snippet

# Calculate Mean Squared Error (MSE)
mse = mean_squared_error(actual_values, predictions)
print('Mean Squared Error (MSE):', mse)

# Calculate R-squared (R2)
r2 = r2_score(actual_values, predictions)
print('R-squared (R2):', r2)

# Plot actual vs predicted values
plt.figure(figsize=(10, 6))
plt.plot(actual_values, label='Actual')
```

```
plt.plot(predictions, label='Predicted')
plt.xlabel('Day')
plt.ylabel('Close Price')
plt.title('Actual vs Predicted Close Price')
plt.legend()
plt.grid(True)
plt.show()
```

```
Mean Squared Error (MSE): 17.63773935566531
R-squared (R2): 0.9977478845961321
```



In [ ]: