In [1]:

```python
import pandas as pd
import numpy as np
#reading the csv file
data=pd.read_csv('iris.csv')
data.head()
```

Out[1]:

	sepallength	sepalwidth	petallength	petalwidth	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [2]:

```python
#shape
data.shape
```

Out[2]:

```
(150, 5)
```

In [3]:

```python
data.describe()
```

Out[3]:

	sepallength	sepalwidth	petallength	petalwidth
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [4]:

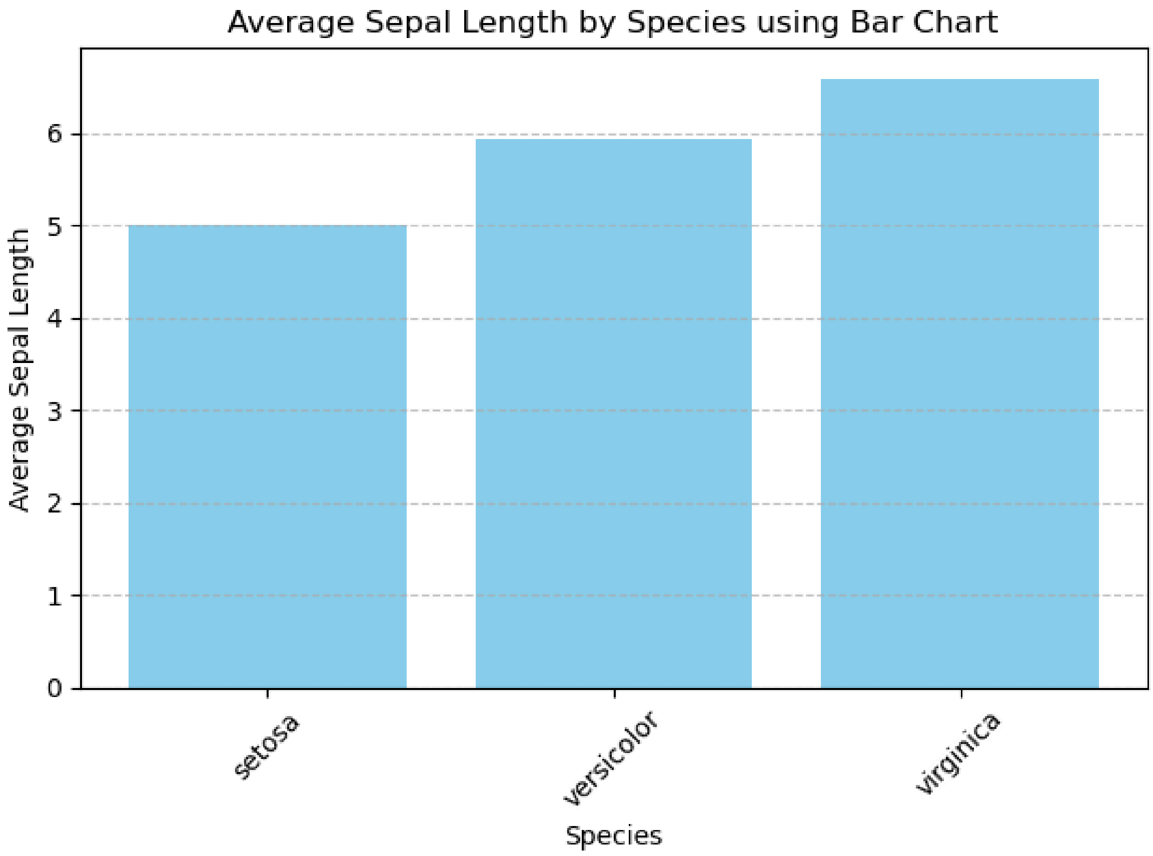
```python
data.value_counts("class")
```

Out[4]:

```
class
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
dtype: int64
```

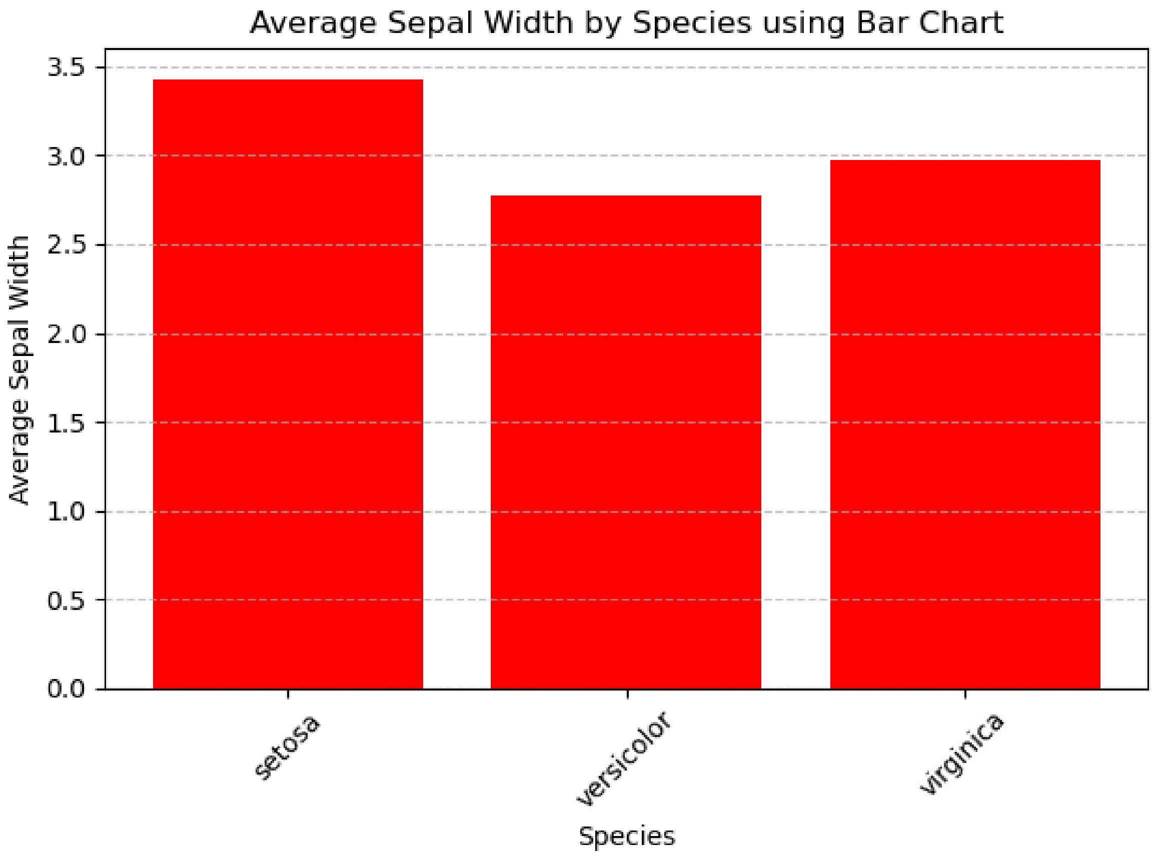
In [21]:

```
import matplotlib.pyplot as plt
import seaborn as sns

# Load the Iris dataset
iris = sns.load_dataset("iris")

# Calculate the average sepal length for each species
average_sepal_length = iris.groupby('species')['sepal_length'].mean()

# Plot the bar chart
plt.bar(average_sepal_length.index, average_sepal_length.values, color='skyblu
plt.title('Average Sepal Length by Species using Bar Chart')
plt.xlabel('Species')
plt.ylabel('Average Sepal Length')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.grid(axis='y', linestyle='--', alpha=0.7)  # Add grid lines for better visi
plt.tight_layout()  # Adjust layout to prevent clipping of labels
plt.show()
```



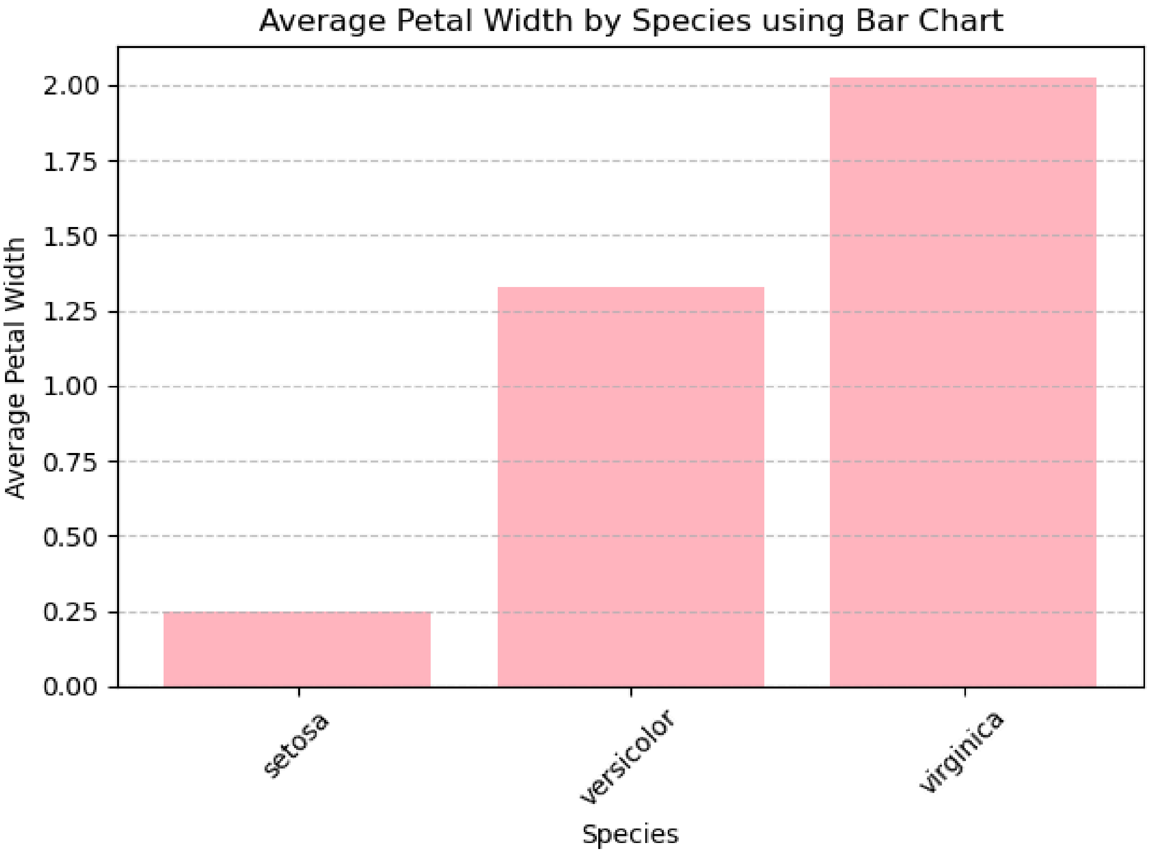
In [24]:

```
import matplotlib.pyplot as plt
import seaborn as sns

# Load the Iris dataset
iris = sns.load_dataset("iris")

# Calculate the average sepal length for each species
average_sepal_width = iris.groupby('species')['sepal_width'].mean()

# Plot the bar chart
plt.bar(average_sepal_width.index, average_sepal_width.values, color='red')
plt.title('Average Sepal Width by Species using Bar Chart')
plt.xlabel('Species')
plt.ylabel('Average Sepal Width')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.grid(axis='y', linestyle='--', alpha=0.7)  # Add grid lines for better visi
plt.tight_layout()  # Adjust layout to prevent clipping of labels
plt.show()
```



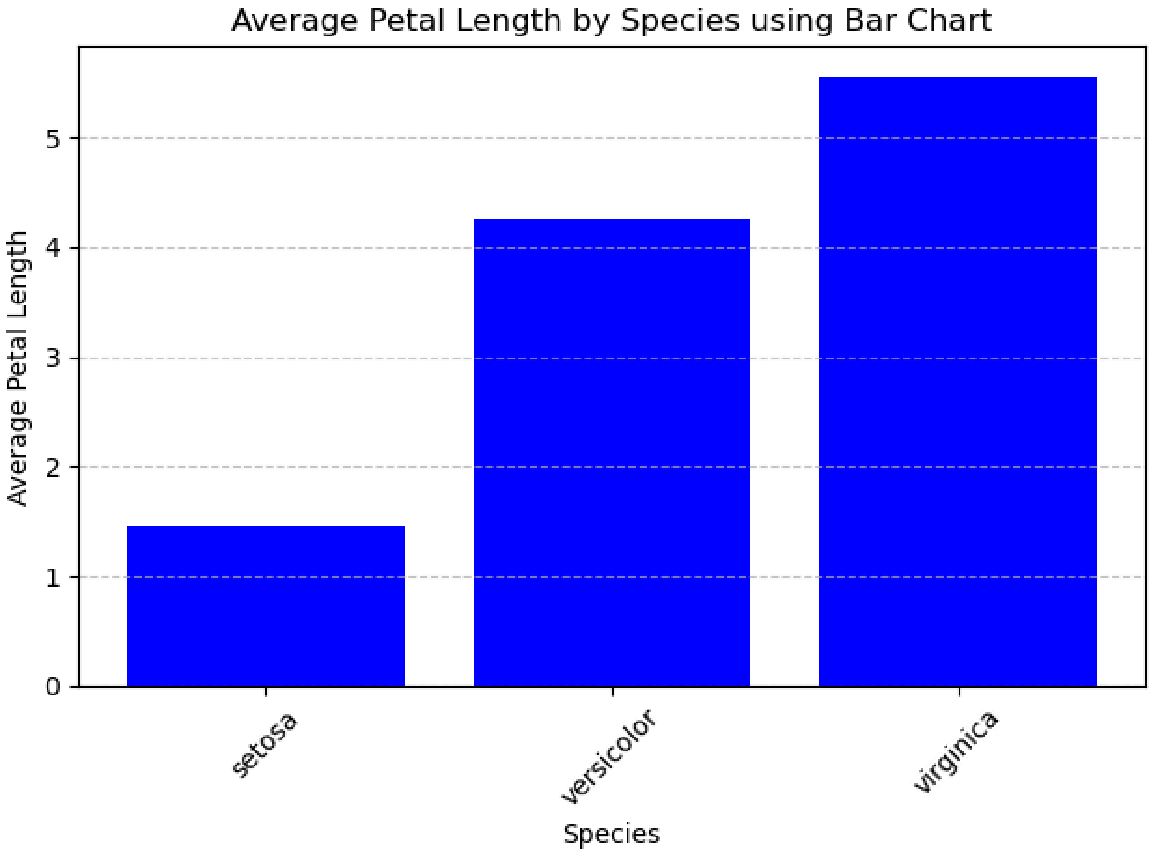
In [26]:

```
import matplotlib.pyplot as plt
import seaborn as sns

# Load the Iris dataset
iris = sns.load_dataset("iris")

# Calculate the average sepal length for each species
average_petal_width = iris.groupby('species')['petal_width'].mean()

# Plot the bar chart
plt.bar(average_petal_width.index, average_petal_width.values, color='lightpin
plt.title('Average Petal Width by Species using Bar Chart')
plt.xlabel('Species')
plt.ylabel('Average Petal Width')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.grid(axis='y', linestyle='--', alpha=0.7)  # Add grid lines for better vis
plt.tight_layout()  # Adjust layout to prevent clipping of labels
plt.show()
```



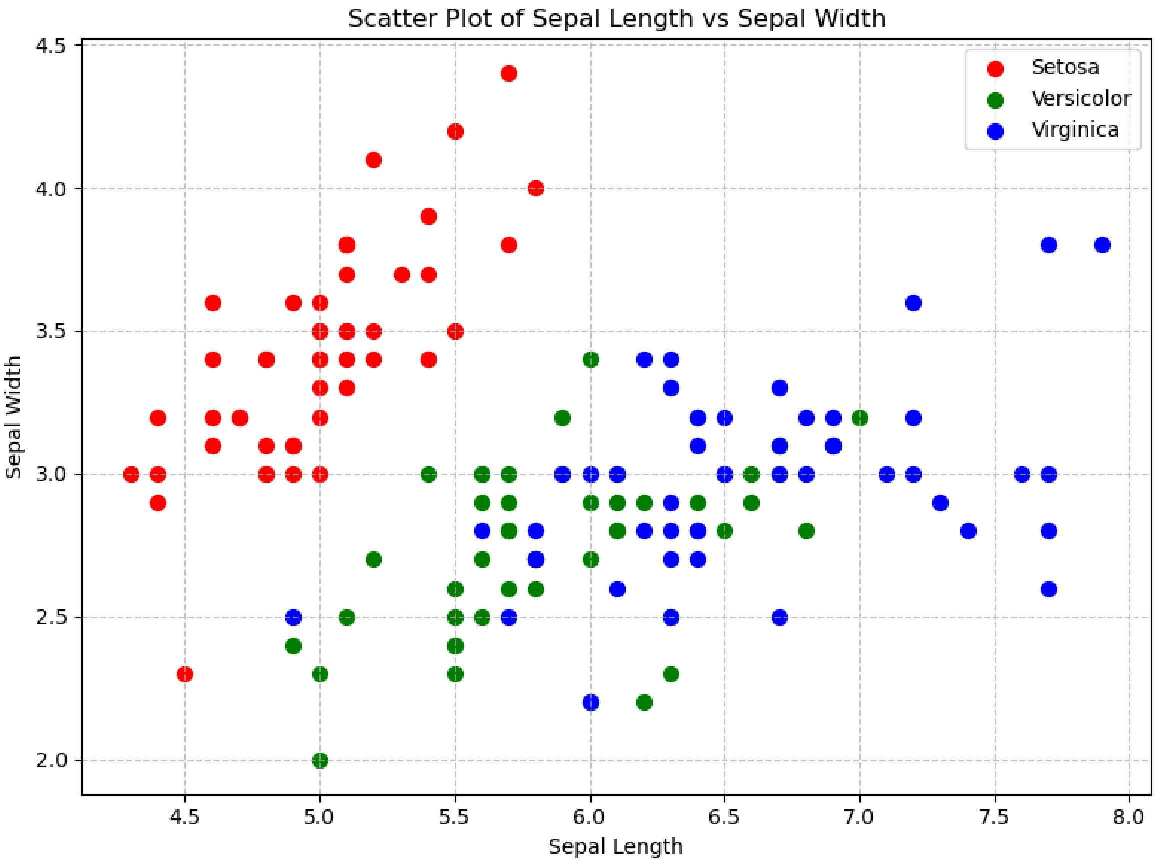
In [27]:

```python
import matplotlib.pyplot as plt
import seaborn as sns

# Load the Iris dataset
iris = sns.load_dataset("iris")

# Calculate the average sepal length for each species
average_petal_length = iris.groupby('species')['petal_length'].mean()

# Plot the bar chart
plt.bar(average_petal_length.index, average_petal_length.values, color='blue')
plt.title('Average Petal Length by Species using Bar Chart')
plt.xlabel('Species')
plt.ylabel('Average Petal Length')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.grid(axis='y', linestyle='--', alpha=0.7)  # Add grid lines for better visu
plt.tight_layout()  # Adjust layout to prevent clipping of labels
plt.show()
```



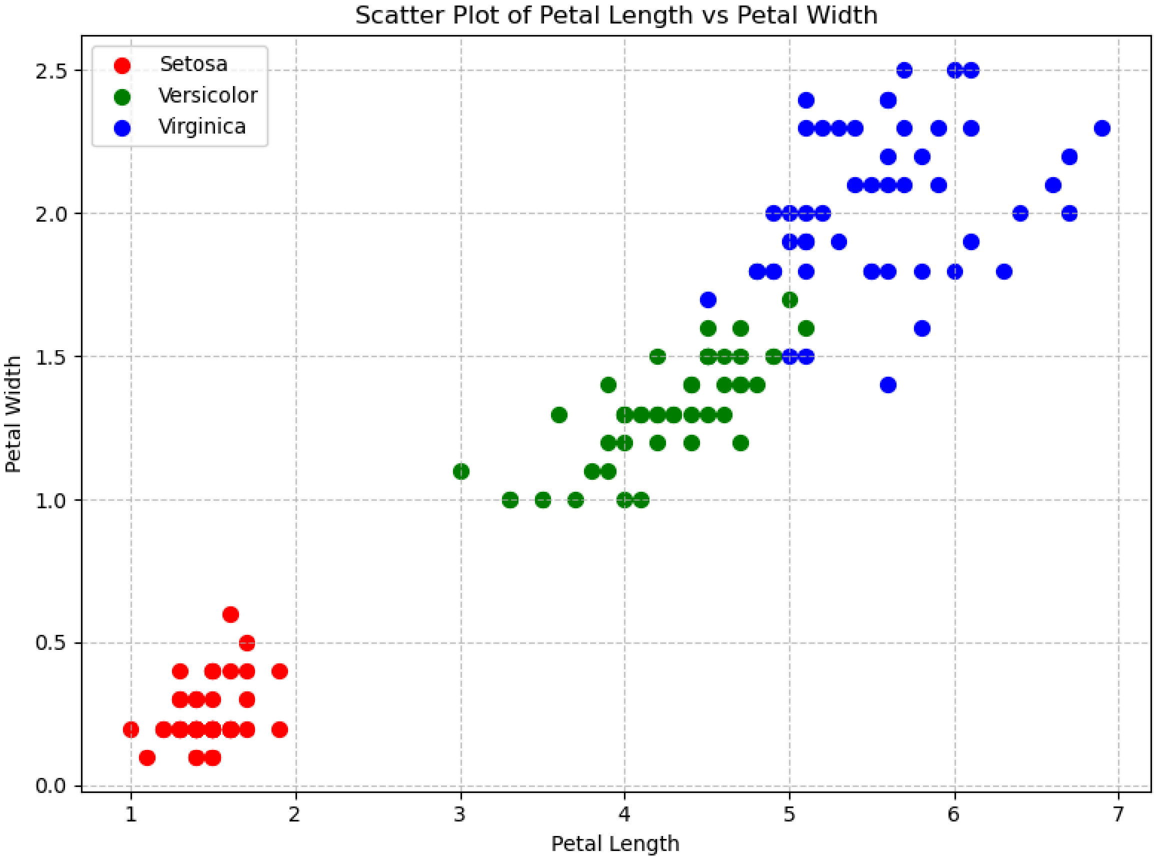
In [22]:

```
import matplotlib.pyplot as plt
import seaborn as sns

# Load the Iris dataset
iris = sns.load_dataset("iris")

# Create separate dataframes for each species
setosa = iris[iris['species'] == 'setosa']
versicolor = iris[iris['species'] == 'versicolor']
virginica = iris[iris['species'] == 'virginica']

# Plot the scatter plot
plt.figure(figsize=(8, 6))

plt.scatter(setosa['sepal_length'], setosa['sepal_width'], label='Setosa', col
plt.scatter(versicolor['sepal_length'], versicolor['sepal_width'], label='Vers
plt.scatter(virginica['sepal_length'], virginica['sepal_width'], label='Virgin

plt.title('Scatter Plot of Sepal Length vs Sepal Width')
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.legend()

plt.grid(True, linestyle='--', alpha=0.7)  # Add grid lines for better visuali
plt.tight_layout()  # Adjust layout to prevent clipping of labels
plt.show()
```



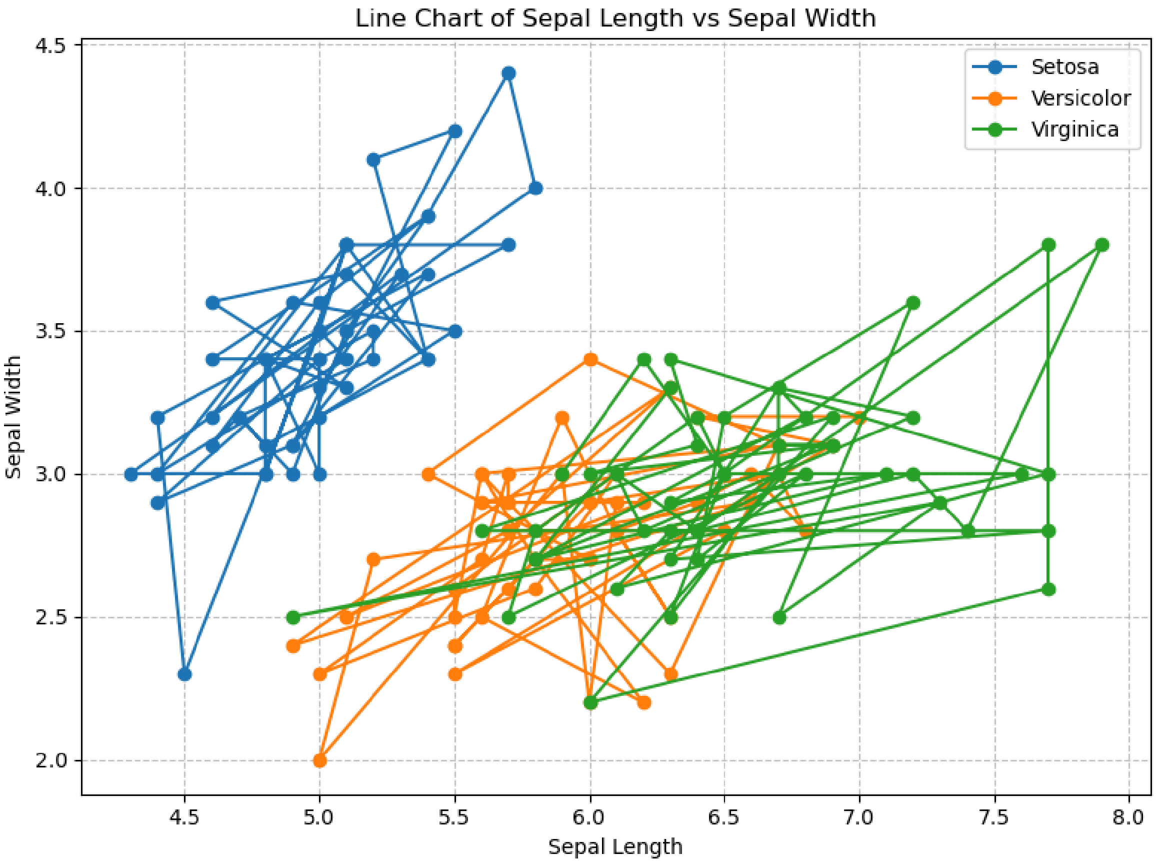
In [29]:

```
import matplotlib.pyplot as plt
import seaborn as sns

# Load the Iris dataset
iris = sns.load_dataset("iris")

# Create separate dataframes for each species
setosa = iris[iris['species'] == 'setosa']
versicolor = iris[iris['species'] == 'versicolor']
virginica = iris[iris['species'] == 'virginica']

# Plot the scatter plot
plt.figure(figsize=(8, 6))

plt.scatter(setosa['petal_length'], setosa['petal_width'], label='Setosa', col
plt.scatter(versicolor['petal_length'], versicolor['petal_width'], label='Vers
plt.scatter(virginica['petal_length'], virginica['petal_width'], label='Virgin

plt.title('Scatter Plot of Petal Length vs Petal Width')
plt.xlabel('Petal Length')
plt.ylabel('Petal Width')
plt.legend()

plt.grid(True, linestyle='--', alpha=0.7)  # Add grid lines for better visuali
plt.tight_layout()  # Adjust layout to prevent clipping of labels
plt.show()
```



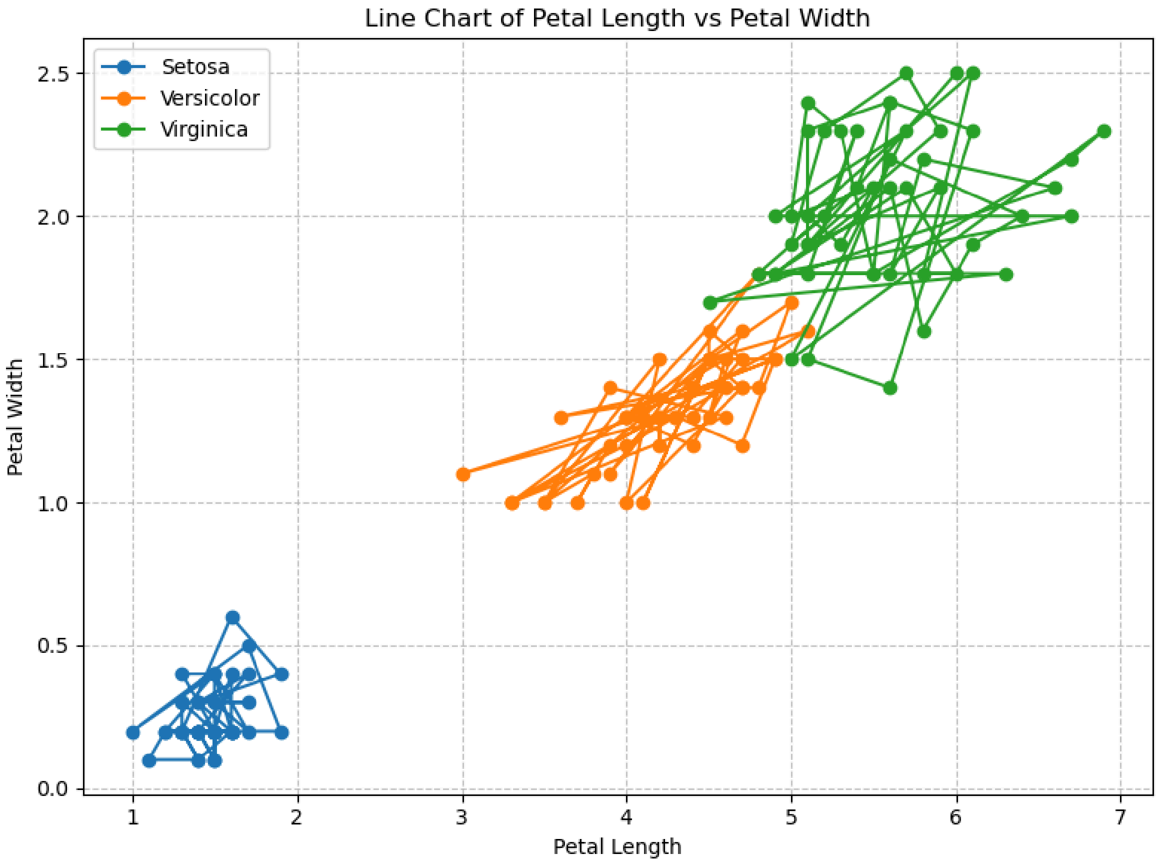
In [23]:

```
import matplotlib.pyplot as plt
import seaborn as sns

# Load the Iris dataset
iris = sns.load_dataset("iris")

# Create separate dataframes for each species
setosa = iris[iris['species'] == 'setosa']
versicolor = iris[iris['species'] == 'versicolor']
virginica = iris[iris['species'] == 'virginica']

# Plot the line chart
plt.figure(figsize=(8, 6))

plt.plot(setosa['sepal_length'], setosa['sepal_width'], label='Setosa', marker
plt.plot(versicolor['sepal_length'], versicolor['sepal_width'], label='Versico
plt.plot(virginica['sepal_length'], virginica['sepal_width'], label='Virginica

plt.title('Line Chart of Sepal Length vs Sepal Width')
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.legend()

plt.grid(True, linestyle='--', alpha=0.7)  # Add grid lines for better visuali
plt.tight_layout()  # Adjust layout to prevent clipping of labels
plt.show()
```



In [28]:

```
import matplotlib.pyplot as plt
import seaborn as sns

# Load the Iris dataset
iris = sns.load_dataset("iris")

# Create separate dataframes for each species
setosa = iris[iris['species'] == 'setosa']
versicolor = iris[iris['species'] == 'versicolor']
virginica = iris[iris['species'] == 'virginica']

# Plot the line chart
plt.figure(figsize=(8, 6))

plt.plot(setosa['petal_length'], setosa['petal_width'], label='Setosa', marker
plt.plot(versicolor['petal_length'], versicolor['petal_width'], label='Versico
plt.plot(virginica['petal_length'], virginica['petal_width'], label='Virginica

plt.title('Line Chart of Petal Length vs Petal Width')
plt.xlabel('Petal Length')
plt.ylabel('Petal Width')
plt.legend()

plt.grid(True, linestyle='--', alpha=0.7)  # Add grid lines for better visuali
plt.tight_layout()  # Adjust layout to prevent clipping of labels
plt.show()
```


Line Chart of Petal Length vs Petal Width
Setosa
Versicolor
Virginica
Petal Width
Petal Length