# IRIS Flower Classification ML Project

TASK 1

In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
from matplotlib.colors import ListedColormap
```

In [7]:

```
# Import the Dataset

data=pd.read_csv("C:/Users/MANOJ S/Downloads/8836201-6f9306ad21398ea43cba4f7d537619
data
```

Out[7]:

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
2	4.7	3.2	1.3	0.2	Setosa
3	4.6	3.1	1.5	0.2	Setosa
4	5.0	3.6	1.4	0.2	Setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Virginica
146	6.3	2.5	5.0	1.9	Virginica
147	6.5	3.0	5.2	2.0	Virginica
148	6.2	3.4	5.4	2.3	Virginica
149	5.9	3.0	5.1	1.8	Virginica

150 rows × 5 columns

In [13]:

```
# Extracting input Featuers and target Variable

x=data[['sepal.length','sepal.width','petal.length','petal.width']].values
y=data['variety'].values
```

In [14]:

```
# converting string values into numeric values using Encoder

from sklearn.preprocessing import LabelEncoder
label_encoder=LabelEncoder()
y=label_encoder.fit_transform(y)
```

In [17]:

```
# Splitting the Dataset into Training and Testing

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
```

In [ ]:

```python
# Train a Logistic Regression Model

from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
model.fit(x_train,y_train)
```

In [27]:

```python
# Make a prediction on Testing Test

y_pred=model.predict(x_test)
y_pred
```

Out[27]:

```
array([1, 0, 2, 1, 1, 0, 1, 2, 1, 1, 2, 0, 0, 0, 0, 1, 2, 1, 1, 2, 0, 2,
       0, 2, 2, 2, 2, 2, 0, 0])
```

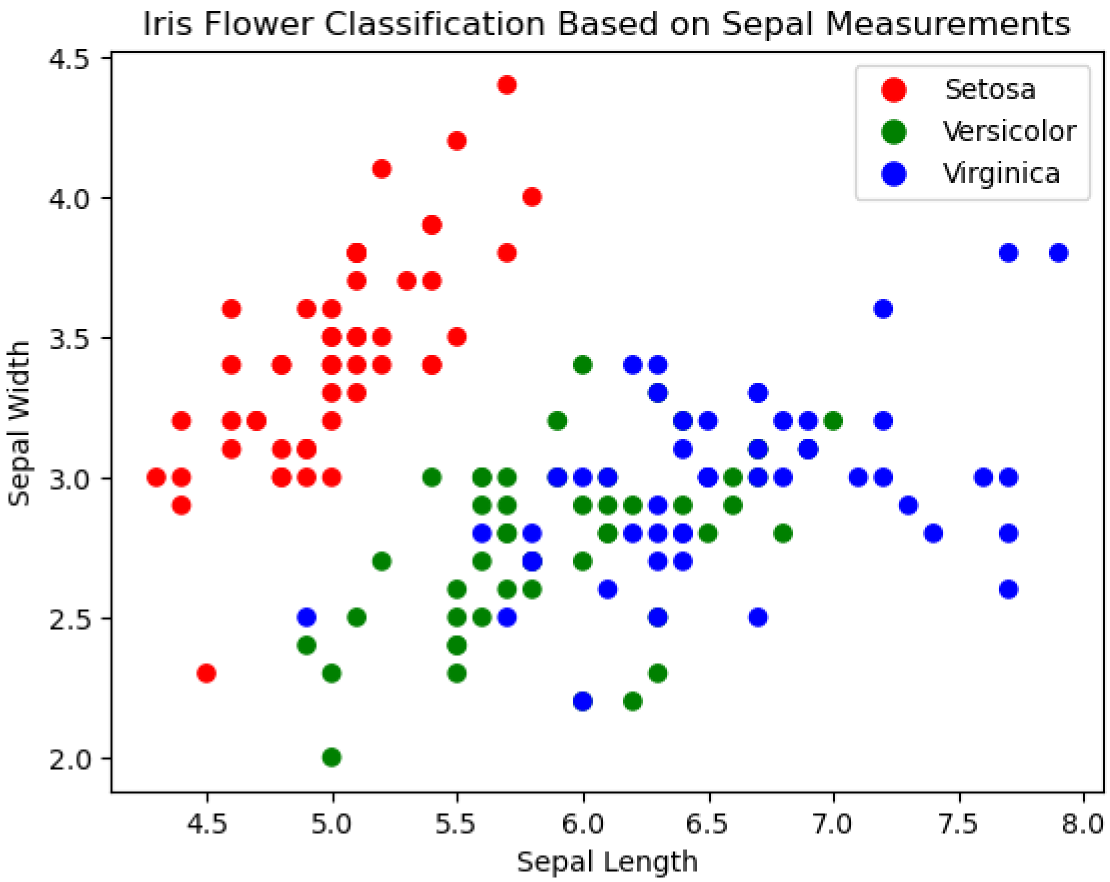
In [20]:

```python
# Evaluate the Model

from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      1.00      1.00         9
           2       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```

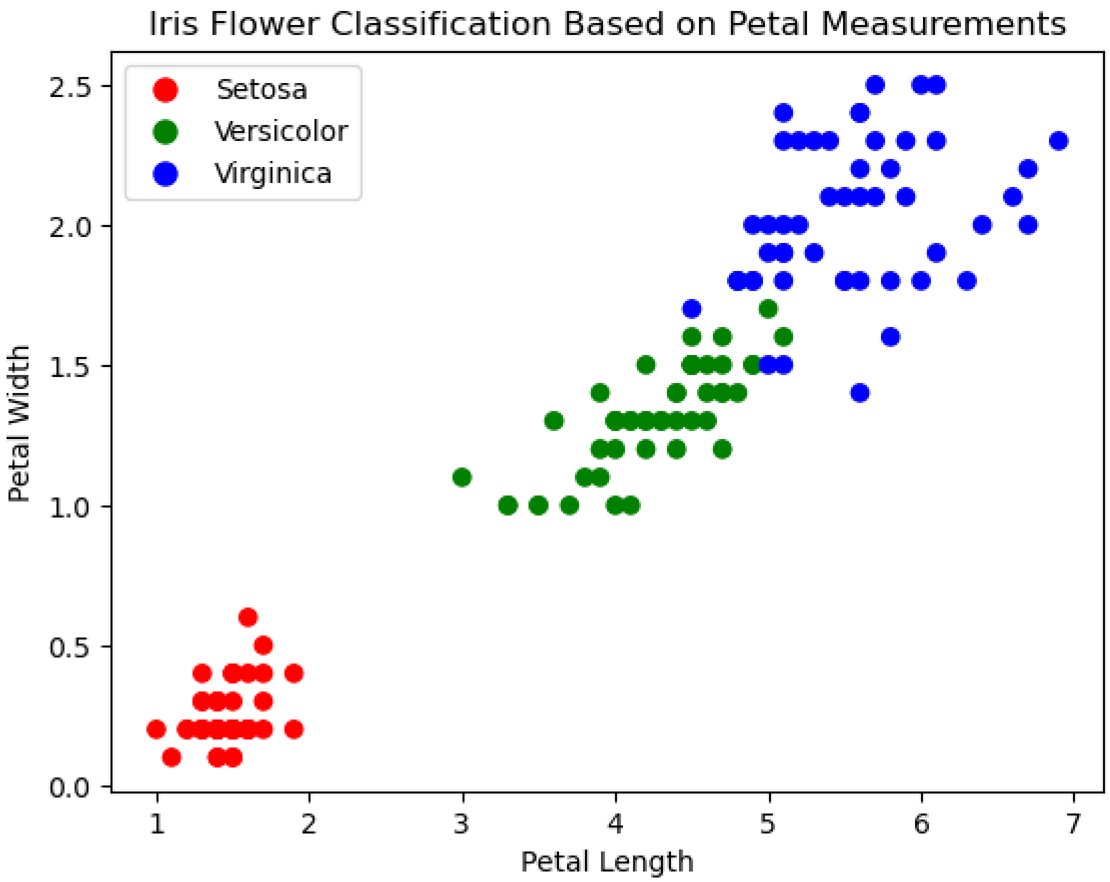
In [21]:

```python
sepal_length=x[:,0]
sepal_width=x[:,1]

class_labels=label_encoder.classes_
class_colors=['red','green','blue']
cmap=ListedColormap(class_colors)

plt.scatter(sepal_length,sepal_width,c=y,cmap=cmap)
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.title('Iris Flower Classification Based on Sepal Measurements')

scatter_handles=[]
for class_label,class_color in zip(class_labels,class_colors):
  scatter_handles.append(plt.Line2D([],[],marker='o',markersize=8,linestyle='',colo
plt.legend(handles=scatter_handles)
plt.show()
```



In [22]:

```
petal_length=x[:,2]
petal_width=x[:,3]

plt.scatter(petal_length,petal_width,c=y,cmap=cmap)
plt.xlabel('Petal Length')
plt.ylabel('Petal Width')
plt.title('Iris Flower Classification Based on Petal Measurements')
scatter_handles=[]
for class_label,class_color in zip(class_labels,class_colors):
  scatter_handles.append(plt.Line2D([],[],marker='o',markersize=8,linestyle='',colo
plt.legend(handles=scatter_handles)
plt.show()
```



In [25]:

```
# Making a Prediction on New Data
import numpy as np

new_data = np.array([[5.1, 3.5, 1.4, 0.2], [6.2, 2.9, 4.3, 1.3], [7.3, 2.8, 6.5, 2.

new_prediction = model.predict(new_data)
print('New Data Predictions: ', new_prediction)

for i, prediction in enumerate(new_prediction):
    print(f'Data point {i + 1}: {label_encoder.inverse_transform([prediction])[0]}'
```

```
New Data Predictions:  [0 1 2]
Data point 1: Setosa
Data point 2: Versicolor
Data point 3: Virginica
```

In [ ]: