In [5]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
data=pd.DataFrame({
    "Sl.No":[1,2,3,4,5,6,7,8,9,10,11,12,
    "Total":[52,71,70,89,79,87,73,67,63,
    "Internal":[30,39,40,47,41,50,38,39,
    "External":[22,32,30,42,38,37,35,28,


})
data.to_csv("5AI REGULAR.csv",index=Fals
print("Dataset loaded from 5AI REGULAR.c
print("Dataset preview")
print(data.head())
numerical_column="Total"
data_num=data[numerical_column]
```

```
Dataset loaded from 5AI REGULAR.csv
Dataset preview
   Sl.No  Total  Internal  External
0      1     52        30        22
1      2     71        39        32
2      3     70        40        30
3      4     89        47        42
4      5     79        41        38
```

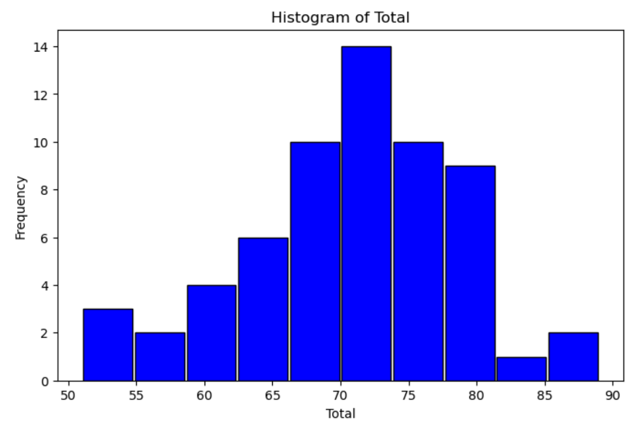
In [6]:

```
mean_val=data_num.mean()
median_val=data_num.median()
mode_val=data_num.mode()
std_dev=data_num.std()
variance=data_num.var()
range_val=data_num.max()-data_num.min()
print(f"Mean:",mean_val)
print(f"Median:",median_val)
print(f"Mode:",mode_val)
print(f"std_dev:",std_dev)
print(f"variance:",variance)
print(f"Range:",range_val)
```

```
Mean: 70.62295081967213
Median: 71.0
Mode: 0    78
Name: Total, dtype: int64
std_dev: 7.922886541377517
variance: 62.772131147540996
Range: 38
```

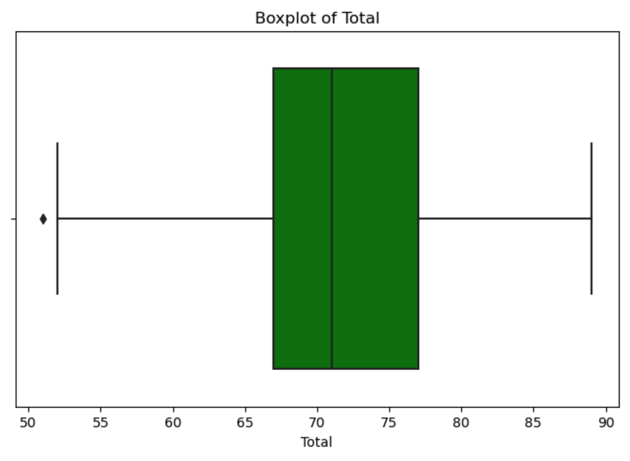
In [9]:

```
plt.figure(figsize=(8,5))
plt.hist(data_num,bins=10,rwidth=0.95,co
plt.title(f"Histogram of {numerical_colu
plt.xlabel(numerical_column)
plt.ylabel("Frequency")
plt.show()
```



In [8]:

```
plt.figure(figsize=(8,5))
sns.boxplot(x=data_num,color="green")
plt.title(f"Boxplot of {numerical_column
plt.show()
```



In [10]:

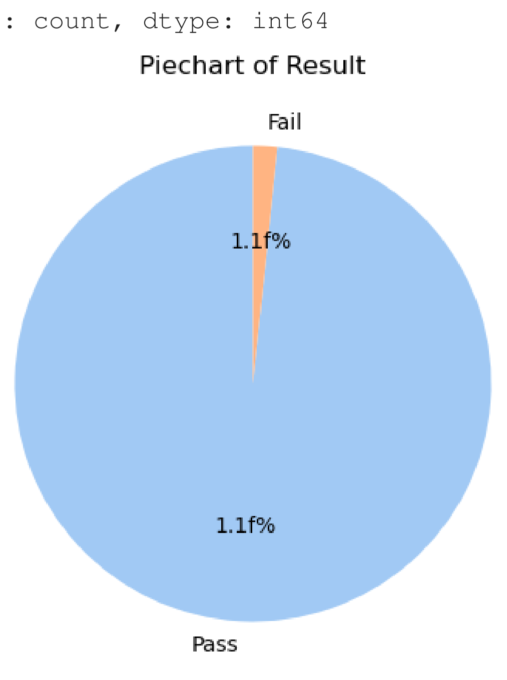
```
q1=data_num.quantile(0.25)
q3=data_num.quantile(0.75)
iqr=q3-q1
lower_bound=q1-1.5*iqr
upper_bound=q3+1.5*iqr
outliers=data_num[(data_num<lower_bound)
print("\n Outliers:")
print(outliers)
```

```
 Outliers:
31    51
Name: Total, dtype: int64
```

In [11]:

```
def total_marks(row):
   if row["Internal"] >= 18 and row["Ext
        return "Pass"
   else:
        return "Fail"
data["Result"]=data.apply(total_marks,ax
categorical_column="Result"
data_cat=data[categorical_column]
category_counter=data_cat.value_counts()
print("\Category freq:")
print(category_counter)
plt.figure(figsize=(8,5))
category_counter.plot(kind="pie",autopct
plt.title(f"Piechart of {categorical_col
plt.ylabel("")
plt.show()
```

```
\Category freq:
Result
Pass    60
Fail     1
Name: count, dtype: int64
```



In [ ]: