In [27]:
```
# Importing Libraries
import pandas as pd
from sklearn.linear_model import LinearRegression as LR
from sklearn.model_selection import train_test_split as TTS
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error as MSE,r2_score as RS,accuracy_
import numpy as np
```

In [2]:
```
# Load Dataset
df=pd.read_csv(r'D:\Student_Marks.csv')
```

In [3]:
```
df.head()
```

Out[3]:

	number_courses	time_study	Marks
0	3	4.508	19.202
1	4	0.096	7.734
2	4	3.133	13.811
3	6	7.909	53.018
4	8	7.811	55.299

In [4]:
```
df.tail()
```

Out[4]:

	number_courses	time_study	Marks
95	6	3.561	19.128
96	3	0.301	5.609
97	4	7.163	41.444
98	7	0.309	12.027
99	3	6.335	32.357

In [5]:
```
df.describe()
```

Out[5]:

	number_courses	time_study	Marks
count	100.000000	100.000000	100.000000
mean	5.290000	4.077140	24.417690
std	1.799523	2.372914	14.326199
min	3.000000	0.096000	5.609000
25%	4.000000	2.058500	12.633000
50%	5.000000	4.022000	20.059500
75%	7.000000	6.179250	36.676250
max	8.000000	7.957000	55.299000

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 3 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   number_courses  100 non-null    int64  
 1   time_study      100 non-null    float64
 2   Marks           100 non-null    float64
dtypes: float64(2), int64(1)
memory usage: 2.5 KB
```

In [7]:
```
df.describe()
```

Out[7]:

	number_courses	time_study	Marks
count	100.000000	100.000000	100.000000
mean	5.290000	4.077140	24.417690
std	1.799523	2.372914	14.326199
min	3.000000	0.096000	5.609000
25%	4.000000	2.058500	12.633000
50%	5.000000	4.022000	20.059500
75%	7.000000	6.179250	36.676250
max	8.000000	7.957000	55.299000

In [8]:
```
# Split The Data To Train And Test
x=df.drop(['Marks'],axis=1)
y=df['Marks']
```

In [12]:
```
x_train,x_test,y_train,y_test=TTS(x,y,test_size=0.2,random_state=10)
```

In [13]:
```
# Fit The Model With LinearRegression
lm=LR()
lm.fit(x_train,y_train)
```

Out[13]:
```
LinearRegression()
```

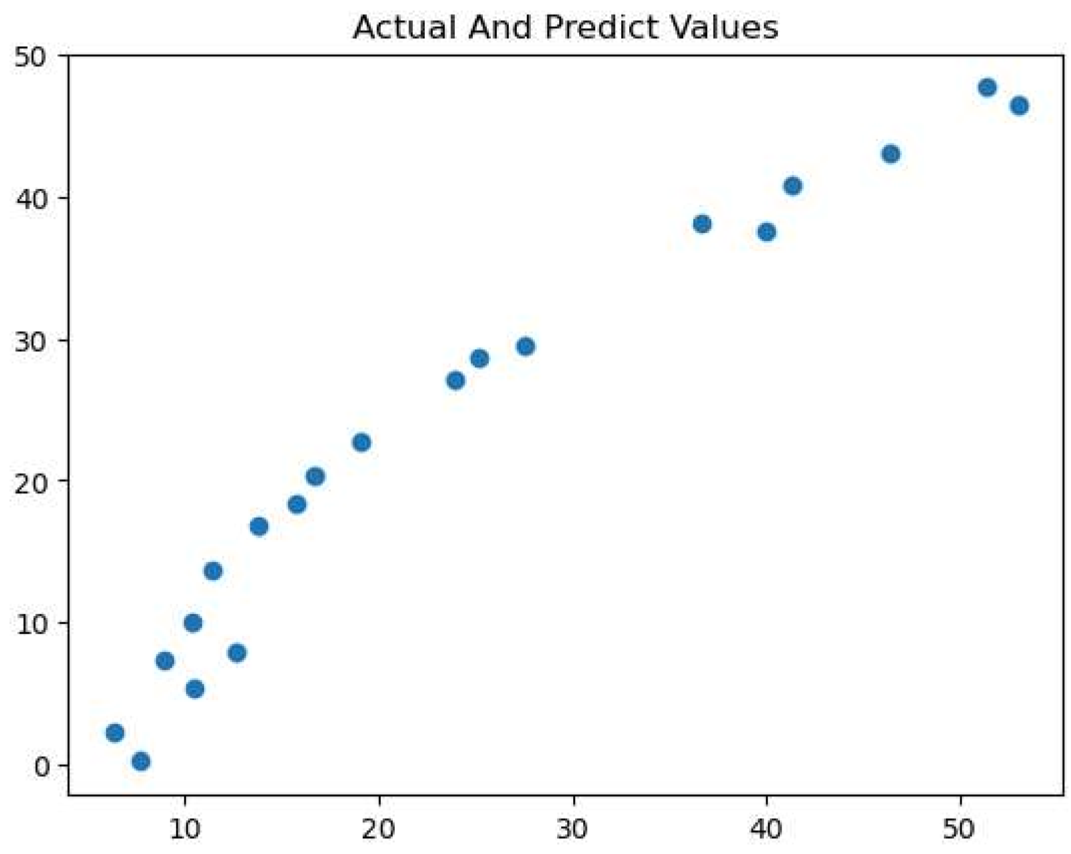
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [14]:
```
y_pred=lm.predict(x_test)
```

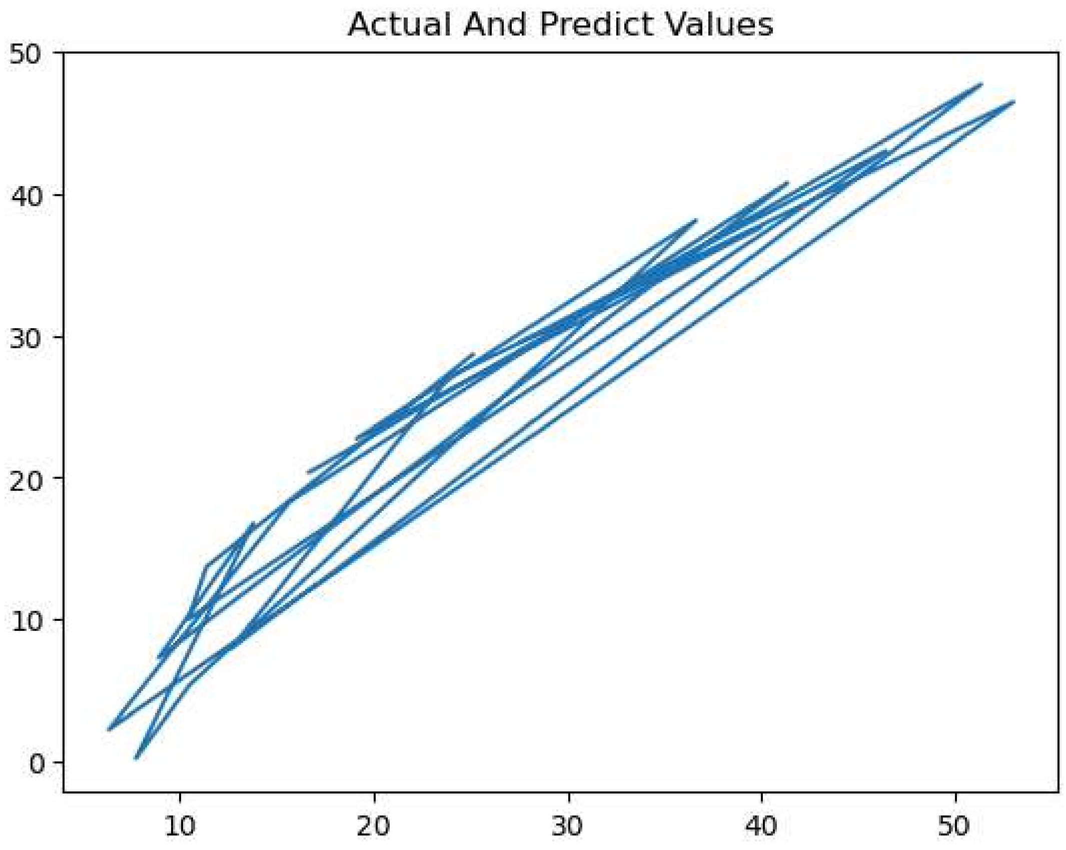
In [18]:
```
# evaluate its performance using metrics like mean squared And R2 Score
mse=MSE(y_test,y_pred)
print(f"Mean Squared error = {mse}")
rs=RS(y_test,y_pred)
print(f"R2 Score = {rs}")
```

```
Mean Squared error = 13.744931448338658
R2 Score = 0.9393161544337274
```

In [22]:
```
# Visualize the regression line and actual vs. predicted values
plt.scatter(y_test,y_pred)
plt.title('Actual And Predict Values')
plt.show()
```



In [21]:
```
plt.plot(y_test,y_pred)
plt.title('Actual And Predict Values')
plt.show()
```



In [30]:
```
# Inserting New Data To Predict
new = np.array([[7,9]])
pred = lm.predict(new)
print(f"Predict New Values = {pred[0]}")
```

```
Predict New Values = 54.20909421097954

C:\Users\nikhil\anaconda3\Lib\site-packages\sklearn\base.py:464: UserWarning:
X does not have valid feature names, but LinearRegression was fitted with fea
ture names
  warnings.warn(
```