IMPORT LIBRARIES

In [82]:

```
import pandas as pd
import numpy as np
```

took dataset from kaggle(csv file)

In [83]:

```
dataset = pd.read_csv(r"C:\Users\khush\Desktop\Salary_Data.csv")
```

to display the csv file

In [84]:

```
dataset.head()
```

Out[84]:

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	Bachelor's	Sales Associate	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0

removed columns which are not required i.e. simplified data

In [85]:

```
columns_to_remove = ['Gender', 'Education Level', 'Job Title']
```

In [86]:

```
dataset = dataset.drop(columns=columns_to_remove)
```

In [87]:

```
dataset.head()
```

Out[87]:

	Age	Years of Experience	Salary
0	32.0	5.0	90000.0
1	28.0	3.0	65000.0
2	45.0	15.0	150000.0
3	36.0	7.0	60000.0
4	52.0	20.0	200000.0

performed various operation on column 'Age'

```
In [88]: dataset.Age.mean()
```

```
Out[88]: 33.62085944494181
```

```
In [89]: dataset.Age.median()
```

```
Out[89]: 32.0
```

```
In [90]: dataset.Age.mode()
```

```
Out[90]: 0    27.0
         Name: Age, dtype: float64
```

```
In [91]: dataset.Age.max()
```

```
Out[91]: 62.0
```

```
In [92]: dataset.Age.min()
```

```
Out[92]: 21.0
```

```
In [93]: column_name = 'Age'
         std_dev = dataset[column_name].std()
```

```
In [94]: print(f"Standard Deviation for {column_name}: {std_dev}")
```

```
Standard Deviation for Age: 7.614632626251171
```

various operation on column 'salary'

```
In [95]: dataset.Salary.mean()
```

```
Out[95]: 115326.96477086132
```

```
In [96]: dataset.Salary.mode()
```

```
Out[96]: 0    140000.0
         Name: Salary, dtype: float64
```

```
In [97]: dataset.Salary.median()
```

```
Out[97]: 115000.0
```

In [98]:

```
dataset.Salary.max()
```

Out[98]:

```
250000.0
```

In [99]:

```
dataset.Salary.min()
```

Out[99]:

```
350.0
```

In [100]:

```
column_name = 'Salary'
std_dev = dataset[column_name].std()
print(f"Standard Deviation for {column_name}: {std_dev}")
```

```
Standard Deviation for Salary: 52786.183910682936
```

operations on entire dataset

In [101]:

```
dataset.mean()
```

Out[101]:

```
Age                        33.620859
Years of Experience         8.094687
Salary                 115326.964771
dtype: float64
```

In [102]:

```
dataset.median()
```

Out[102]:

```
Age                        32.0
Years of Experience         7.0
Salary                 115000.0
dtype: float64
```

In [103]:

```
dataset.min()
```

Out[103]:

```
Age                     21.0
Years of Experience      0.0
Salary                 350.0
dtype: float64
```

In [104]:

```
dataset.max()
```

Out[104]:

```
Age                        62.0
Years of Experience        34.0
Salary                 250000.0
dtype: float64
```

In [105]:

```
dataset.Age.mode()
```

Out[105]:

```
0    27.0
Name: Age, dtype: float64
```

```
In [106]: dataset.groupby(['Age']).count()
```

Out[106]:

	Years of Experience	Salary
Age		
21.0	18	18
22.0	15	15
23.0	104	104
24.0	240	240
25.0	284	284
26.0	393	393
27.0	517	517
28.0	429	429
29.0	444	444
30.0	449	449
31.0	365	364
32.0	351	351
33.0	398	398
34.0	309	309
35.0	200	200
36.0	282	281
37.0	156	156
38.0	149	149
39.0	158	158
40.0	92	92
41.0	129	129
42.0	176	176
43.0	158	158
44.0	126	126
45.0	144	144
46.0	102	102
47.0	47	47
48.0	98	98
49.0	91	91
50.0	88	88
51.0	30	30
52.0	29	29
53.0	7	7
54.0	68	68
55.0	16	16

	Years of Experience	Salary
Age		
56.0	11	11
57.0	9	9
58.0	7	7
60.0	5	5
61.0	2	2
62.0	5	5

In [107]:

```
dataset.isnull()
```

Out[107]:

	Age	Years of Experience	Salary
0	False	False	False
1	False	False	False
2	False	False	False
3	False	False	False
4	False	False	False
...	...	...	...
6699	False	False	False
6700	False	False	False
6701	False	False	False
6702	False	False	False
6703	False	False	False

6704 rows × 3 columns

In [108]:

```
std_dev = np.std(dataset)
```

In [109]:

```
print("standard deviation:", std_dev)
```

```
standard deviation: Age                        7.614065
Years of Experience        6.058551
Salary                 52782.243908
dtype: float64
```

Box Plot

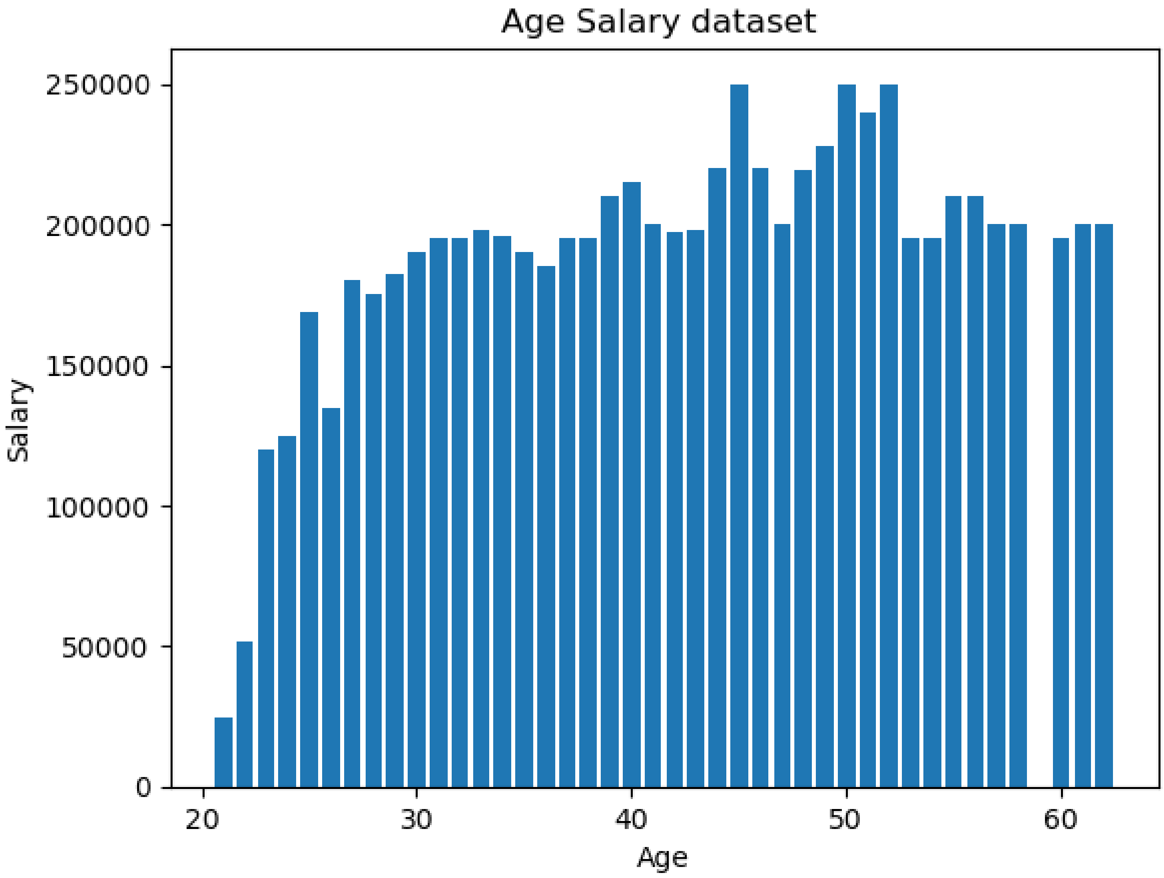
In [110]:

```
import matplotlib.pyplot as plt
```

```
In [111]: dataset = pd.read_csv(r"C:\Users\khush\Desktop\Salary_Data.csv")
```

```
In [112]: x = dataset['Age']
          y = dataset['Salary']
```

```
In [113]: plt.title('Age Salary dataset ')
          plt.xlabel('Age')
          plt.ylabel('Salary')
          plt.bar(x , y)
          plt.show()
```



interquartile range

```
In [114]: first_quartile = np.percentile(Age, 25)
```

```
In [115]: print("25% :",first_quartile)
```

```
25% : -0.6727350412768447
```

```
In [116]: third_quartile = np.percentile(Age, 75)
```

In [117]:

```
print("75% :", third_quartile)
```

```
75% : 0.6434662147791437
```

In [118]:

```
first_quartile = np.percentile(Age, 25)
third_quartile = np.percentile(Age, 75)
iqr = third_quartile - first_quartile
```

In [119]:

```
print("Interquartile Range (IQR):", iqr)
```

```
Interquartile Range (IQR): 1.3162012560559884
```

summary stats function

In [120]:

```
summary_stats = dataset.groupby('Age')['Salary'].describe()
```

```
In [121]: print(summary_stats)
```

```
      count           mean           std       min        25%       50%  \
Age                                                                         
21.0   18.0   25000.000000      0.000000   25000.0   25000.00   25000.0   
22.0   15.0   32910.933333  11880.865333   25000.0   25000.00   25000.0   
23.0  104.0   47192.009615  18035.442395     579.0   35000.00   50000.0   
24.0  240.0   51052.250000  26124.985328   25000.0   25000.00   48132.5   
25.0  284.0   63730.387324  28777.491135     550.0   35000.00   60000.0   
26.0  393.0   65949.106870  31167.205609   26000.0   45000.00   55000.0   
27.0  517.0   77432.729207  33012.912794   30000.0   55000.00   70000.0   
28.0  429.0   84140.368298  40685.862170   25000.0   60000.00   70000.0   
29.0  444.0   85764.750000  40882.104745     350.0   60000.00   75000.0   
30.0  449.0  100903.955457  44141.603325   25000.0   65000.00   95000.0   
31.0  364.0  114381.461538  49219.650931     500.0   70000.00  120000.0   
32.0  351.0  120500.179487  40343.238208   40000.0   90000.00  120000.0   
33.0  398.0  127425.266332  41148.195089   25000.0  100000.00  123520.5   
34.0  309.0  132065.291262  38029.021237   50000.0  100000.00  140000.0   
35.0  200.0  119365.835000  28282.573449   35000.0   95000.00  120000.0   
36.0  281.0  134071.725979  35196.096365   35000.0  100000.00  139413.0   
37.0  156.0  146939.461538  32923.368331   75000.0  115000.00  150000.0   
38.0  149.0  138185.570470  30591.206601   80000.0  119000.00  136449.0   
39.0  158.0  147282.715190  37107.978657   55000.0  120000.00  145000.0   
40.0   92.0  149862.989130  38457.087535   60000.0  122000.00  150000.0   
41.0  129.0  155235.565891  32816.117854   80000.0  131000.00  151315.0   
42.0  176.0  155952.477273  23802.443693  100000.0  139571.50  170000.0   
43.0  158.0  165409.398734  19730.539525  105000.0  150000.00  162000.0   
44.0  126.0  160518.873016  18377.397914  110000.0  150000.00  162037.5   
45.0  144.0  170334.291667  26086.731136   80000.0  150000.00  180000.0   
46.0  102.0  171242.362745  16885.301897  120000.0  160000.00  180000.0   
47.0   47.0  178750.106383  11582.239177  135000.0  170000.00  178859.0   
48.0   98.0  192723.846939  16884.955629  140000.0  180255.25  190000.0   
49.0   91.0  189013.582418  18146.688329  120177.0  182392.00  185000.0   
50.0   88.0  190849.897727  18832.132979  130000.0  180000.00  190000.0   
51.0   30.0  200356.066667  27080.013909  130000.0  190000.00  190000.0   
52.0   29.0  187629.379310  15720.313915  161568.0  185462.00  186963.0   
53.0    7.0  180806.571429  11812.715266  166109.0  173054.50  181714.0   
54.0   68.0  189816.250000   8875.741981  158254.0  190000.00  192103.5   
55.0   16.0  198482.187500  10884.112294  183020.0  190407.25  193964.0   
56.0   11.0  196577.636364   4507.360808  195000.0  195000.00  195000.0   
57.0    9.0  176993.777778  31698.597168  121450.0  188232.00  191790.0   
58.0    7.0  195715.428571   4496.983539  190004.0  192502.00  195000.0   
60.0    5.0  186132.400000   6855.522467  179180.0  179180.00  188651.0   
61.0    2.0  200000.000000      0.000000  200000.0  200000.00  200000.0   
62.0    5.0  200000.000000      0.000000  200000.0  200000.00  200000.0   

           75%       max  
Age                       
21.0   25000.0   25000.0  
22.0   45000.0   51832.0  
23.0   52807.0  119836.0  
24.0   60000.0  125000.0  
25.0   90000.0  169159.0  
26.0   85000.0  135000.0  
27.0   80000.0  180000.0  
28.0  110000.0  175000.0  
29.0   95000.0  182000.0  
30.0  120000.0  190000.0  
31.0  140000.0  195000.0  
```

```
32.0  145000.0  195000.0
33.0  148000.0  198000.0
34.0  160976.0  196000.0
35.0  140000.0  190000.0
36.0  160000.0  185000.0
37.0  170000.0  195000.0
38.0  155000.0  195000.0
39.0  170000.0  210000.0
40.0  160000.0  215000.0
41.0  185000.0  200000.0
42.0  180000.0  197000.0
43.0  185000.0  198000.0
44.0  170000.0  220000.0
45.0  185000.0  250000.0
46.0  180000.0  220000.0
47.0  190000.0  200000.0
48.0  210000.0  219000.0
49.0  195000.0  228000.0
50.0  200000.0  250000.0
51.0  230000.0  240000.0
52.0  190596.0  250000.0
53.0  188357.0  195000.0
54.0  195000.0  195270.0
55.0  210000.0  210000.0
56.0  195000.0  210000.0
57.0  195000.0  200000.0
58.0  200000.0  200000.0
60.0  188651.0  195000.0
61.0  200000.0  200000.0
62.0  200000.0  200000.0
```

```
In [ ]:
```