Importing the Required Libraries

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

Load the Dataset

In [2]:

```
df=pd.read_csv('Salary_Prediction_project.csv')
df.head()
```

Out[2]:

	rank	discipline	yrs.since.phd	yrs.service	sex	salary
0	Prof	B	19	18	Male	139750
1	Prof	B	20	16	Male	173200
2	AsstProf	B	4	3	Male	79750
3	Prof	B	45	39	Male	115000
4	Prof	B	40	41	Male	141500

In [3]:

```
#check the row and column
df.shape
```

Out[3]:

```
(397, 6)
```

In [4]:

```
#check the information the dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 397 entries, 0 to 396
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   rank           397 non-null    object
 1   discipline     397 non-null    object
 2   yrs.since.phd  397 non-null    int64 
 3   yrs.service    397 non-null    int64 
 4   sex            397 non-null    object
 5   salary         397 non-null    int64 
dtypes: int64(3), object(3)
memory usage: 18.7+ KB
```

In [5]:

```
#check the missing values
df.isnull().sum()
```

Out[5]:

```
rank             0
discipline       0
yrs.since.phd    0
yrs.service      0
sex              0
salary           0
dtype: int64
```

In [6]:

```
#check the columns
df.dtypes
```

Out[6]:

```
rank             object
discipline       object
yrs.since.phd     int64
yrs.service       int64
sex              object
salary            int64
dtype: object
```

In [7]:

```
df.describe()
```

Out[7]:

	yrs.since.phd	yrs.service	salary
count	397.000000	397.000000	397.000000
mean	22.314861	17.614610	113706.458438
std	12.887003	13.006024	30289.038695
min	1.000000	0.000000	57800.000000
25%	12.000000	7.000000	91000.000000
50%	21.000000	16.000000	107300.000000
75%	32.000000	27.000000	134185.000000
max	56.000000	60.000000	231545.000000

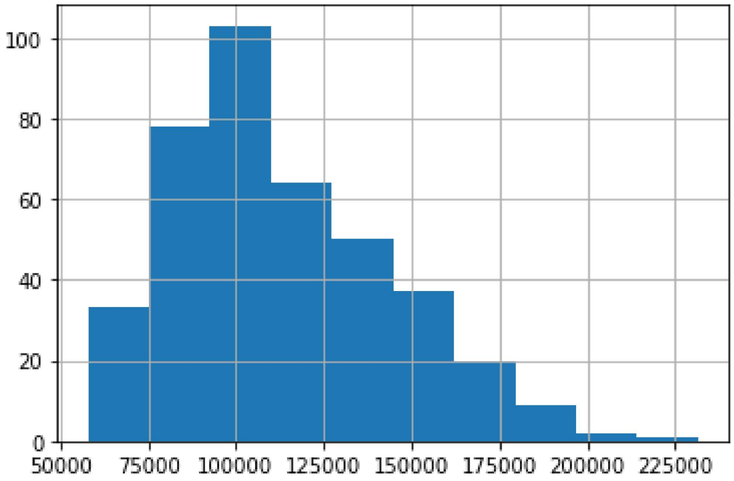
Data visualization

In [8]:

```
df.salary.hist()
```

Out[8]:

```
<AxesSubplot:>
```

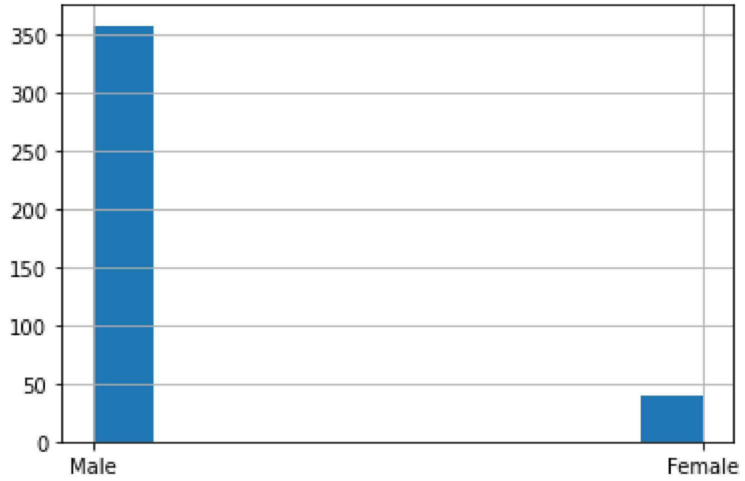


In [9]:

```
df.sex.hist()
```

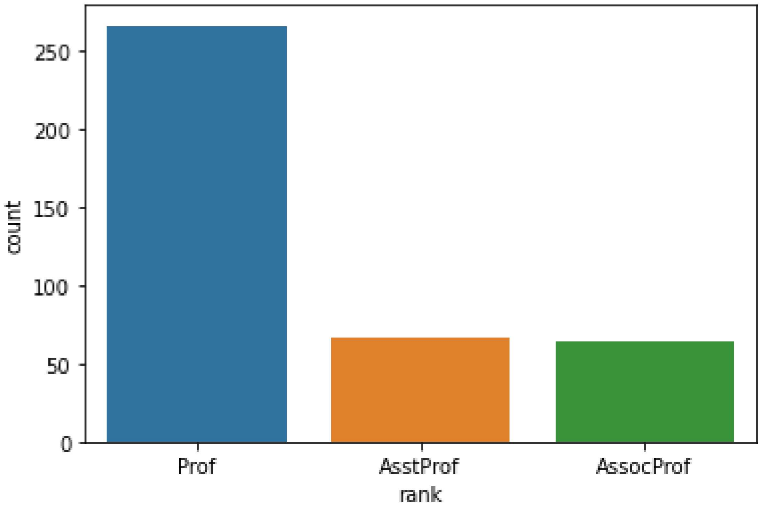
Out[9]:

```
<AxesSubplot:>
```



In [10]:
```
sns.countplot(x='rank',data=df)
print(df['rank'].value_counts())
```

```
Prof         266
AsstProf      67
AssocProf     64
Name: rank, dtype: int64
```



In [11]:
```
df.isnull().sum()
```

Out[11]:
```
rank             0
discipline       0
yrs.since.phd    0
yrs.service      0
sex              0
salary           0
dtype: int64
```

In [12]:
```
df.shape
```

Out[12]:
```
(397, 6)
```

In [13]:
```
df.dropna(how='any').shape
```

Out[13]:
```
(397, 6)
```

In [14]:
```
df.isnull().sum()
```

Out[14]:

```
rank             0
discipline       0
yrs.since.phd    0
yrs.service      0
sex              0
salary           0
dtype: int64
```

In [15]:

```
df["rank"].unique()
```

Out[15]:

```
array(['Prof', 'AsstProf', 'AssocProf'], dtype=object)
```

In [16]:

```
df["rank"].unique().size
```

Out[16]:

```
3
```

Handling Categorical Features or Columns

In [17]:

```
df.head()
```

Out[17]:

	rank	discipline	yrs.since.phd	yrs.service	sex	salary
0	Prof	B	19	18	Male	139750
1	Prof	B	20	16	Male	173200
2	AsstProf	B	4	3	Male	79750
3	Prof	B	45	39	Male	115000
4	Prof	B	40	41	Male	141500

In [18]:

```
#Handling categorical feature ---> Gender
df['sex']=df['sex'].map({'Female':0,'Male':1})
df.head()
```

Out[18]:

	rank	discipline	yrs.since.phd	yrs.service	sex	salary
0	Prof	B	19	18	1	139750
1	Prof	B	20	16	1	173200
2	AsstProf	B	4	3	1	79750
3	Prof	B	45	39	1	115000
4	Prof	B	40	41	1	141500

In [19]:

```
df['rank'].unique()
```

Out[19]:

```
array(['Prof', 'AsstProf', 'AssocProf'], dtype=object)
```

In [20]:

```
#Handling categorical feature ---> rank
df['rank']=df['rank'].map({'Prof':0,'AsstProf':1,'AssoProf':2})
df.head()
```

```
Out[20]:
```

	rank	discipline	yrs.since.phd	yrs.service	sex	salary
0	0.0	B	19	18	1	139750
1	0.0	B	20	16	1	173200
2	1.0	B	4	3	1	79750
3	0.0	B	45	39	1	115000
4	0.0	B	40	41	1	141500

```
In [21]: df['discipline'].unique()

Out[21]: array(['B', 'A'], dtype=object)
```

```
In [22]: df['discipline']=df['discipline'].map({'B':0,'A':1})
         df.head()
```

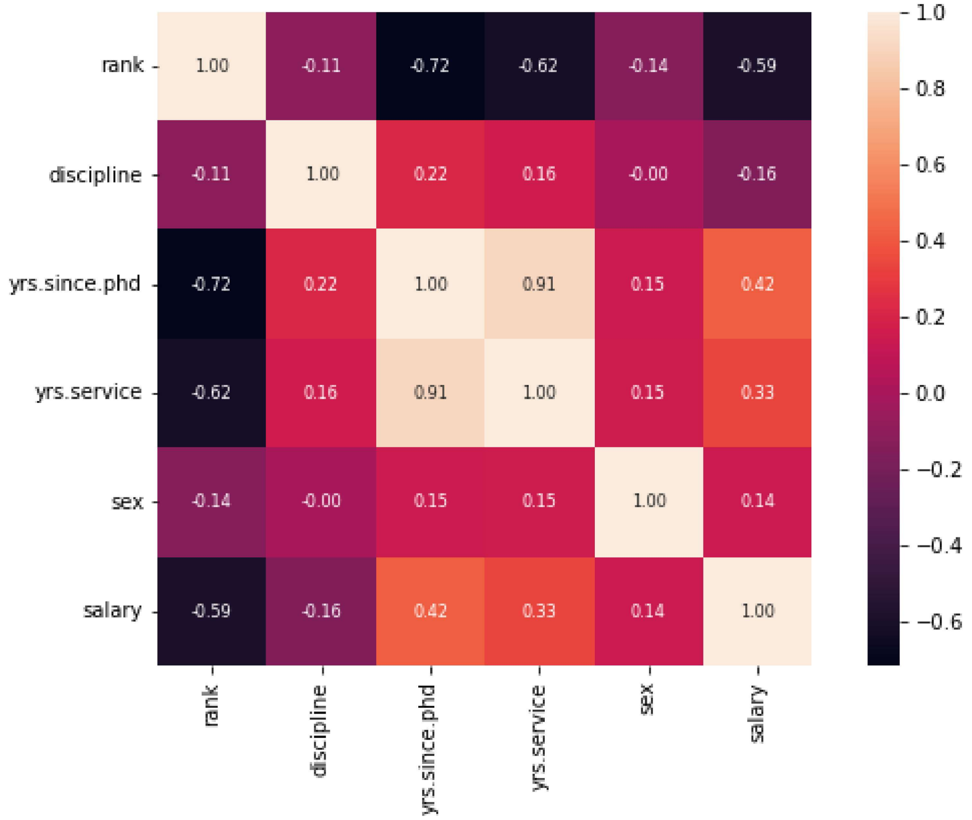
```
Out[22]:
```

	rank	discipline	yrs.since.phd	yrs.service	sex	salary
0	0.0	0	19	18	1	139750
1	0.0	0	20	16	1	173200
2	1.0	0	4	3	1	79750
3	0.0	0	45	39	1	115000
4	0.0	0	40	41	1	141500

Correlation Heat Map

```
In [23]: plt.figure(figsize=(10,6))
         sns.heatmap(df.corr(),cbar=True,square=True,fmt='.2f',annot=True,annot_kws={'size':8})

Out[23]: <AxesSubplot:>
```



In [ ]:

```
```

In [24]:

```
X = df.iloc[:, :-1].values
y = df.iloc[:,1:].values
```

In [25]:

```
X.shape
y.shape
```

Out[25]:

```
(397, 5)
```

In [32]:

```
# Split data 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=0.3, random_state=
```

In [33]:

```
print(X.shape)
print(y.shape)
```

```
(442, 10)
(442,)
```

In [34]:

```
 from sklearn.ensemble import HistGradientBoostingRegressor
from sklearn.datasets import load_diabetes
 X, y = load_diabetes(return_X_y=True)
est = HistGradientBoostingRegressor().fit(X, y)
 est.score(X, y)
```

Out[34]:

```
0.9299589575098558
```