Import dataset

In [1]:
```
import pandas as pd
df=pd.read_csv("C:/Users/FamiAmal/Downloads/Employee.csv")
df
```

Out[1]:

	Company	Age	Salary	Place	Country	Gender
0	TCS	20.0	NaN	Chennai	India	0
1	Infosys	30.0	NaN	Mumbai	India	0
2	TCS	35.0	2300.0	Calcutta	India	0
3	Infosys	40.0	3000.0	Delhi	India	0
4	TCS	23.0	4000.0	Mumbai	India	0
...	...	...	...	...	...	...
143	TCS	33.0	9024.0	Calcutta	India	1
144	Infosys	22.0	8787.0	Calcutta	India	1
145	Infosys	44.0	4034.0	Delhi	India	1
146	TCS	33.0	5034.0	Mumbai	India	1
147	Infosys	22.0	8202.0	Cochin	India	0

148 rows × 6 columns

In [2]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 148 entries, 0 to 147
Data columns (total 6 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   Company  140 non-null    object 
 1   Age      130 non-null    float64
 2   Salary   124 non-null    float64
 3   Place    134 non-null    object 
 4   Country  148 non-null    object 
 5   Gender   148 non-null    int64  
dtypes: float64(2), int64(1), object(3)
memory usage: 7.1+ KB
```

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

Out[3]:

	Company	Age	Salary	Place	Country	Gender
0	TCS	20.0	NaN	Chennai	India	0
1	Infosys	30.0	NaN	Mumbai	India	0
2	TCS	35.0	2300.0	Calcutta	India	0
3	Infosys	40.0	3000.0	Delhi	India	0
4	TCS	23.0	4000.0	Mumbai	India	0

In [4]:
```
df.isnull().sum()
```

Out[4]:
```
Company     8
Age        18
Salary     24
Place      14
Country     0
Gender      0
dtype: int64
```

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

Out[5]:

	Age	Salary	Gender
count	130.000000	124.000000	148.000000
mean	30.484615	5312.467742	0.222973
std	11.096640	2573.764683	0.417654
min	0.000000	1089.000000	0.000000
25%	22.000000	3030.000000	0.000000
50%	32.500000	5000.000000	0.000000
75%	37.750000	8000.000000	0.000000
max	54.000000	9876.000000	1.000000

In [6]:
```
df['Company'].value_counts()
```

Out[6]:
```
Company
TCS                           53
Infosys                       45
CTS                           36
Tata Consultancy Services      2
Congnizant                     2
Infosys Pvt Lmt                2
Name: count, dtype: int64
```

In [7]:
```
df['Place'].value_counts()
```

Out[7]:
```
Place
Mumbai        37
Calcutta      33
Chennai       14
Delhi         14
Cochin        13
Noida          8
Hyderabad      8
Podicherry     3
Pune           2
Bhopal         1
Nagpur         1
Name: count, dtype: int64
```

In [8]:
```
df['Country'].value_counts()
```

Out[8]:
```
Country
India    148
Name: count, dtype: int64
```

In [9]:
```
df['Gender'].value_counts()
```

Out[9]:
```
Gender
0    115
1     33
Name: count, dtype: int64
```

In [10]:
```
df1=df.rename({'Company':'Comp_name','Age':'Emp_age','Salary':'Emp_salary'
df1
```

Out[10]:

	Comp_name	Emp_age	Emp_salary	Emp_place	Emp_country	Emp_gender
0	TCS	20.0	NaN	Chennai	India	0
1	Infosys	30.0	NaN	Mumbai	India	0
2	TCS	35.0	2300.0	Calcutta	India	0
3	Infosys	40.0	3000.0	Delhi	India	0
4	TCS	23.0	4000.0	Mumbai	India	0
...	...	...	...	...	...	...
143	TCS	33.0	9024.0	Calcutta	India	1
144	Infosys	22.0	8787.0	Calcutta	India	1
145	Infosys	44.0	4034.0	Delhi	India	1
146	TCS	33.0	5034.0	Mumbai	India	1
147	Infosys	22.0	8202.0	Cochin	India	0

148 rows × 6 columns

In [11]:
```
df1.shape
```

Out[11]:
```
(148, 6)
```

Data Cleaning

In [12]:
```
df1.duplicated().sum()
```

Out[12]:
```
4
```

In [13]:
```
df1.drop_duplicates(inplace=True)
df1.shape[0]
```

Out[13]:
```
144
```

In [14]:
```
df1.isnull().sum()
```

Out[14]:
```
Comp_name       8
Emp_age        17
Emp_salary     23
Emp_place      14
Emp_country     0
Emp_gender      0
dtype: int64
```

In [15]:
```
round(df1.isnull().mean()*100,2)
```

Out[15]:
```
Comp_name       5.56
Emp_age        11.81
Emp_salary     15.97
Emp_place       9.72
Emp_country     0.00
Emp_gender      0.00
dtype: float64
```

In [16]:
```
df1.dropna(subset=['Comp_name'],axis=0,inplace=True)
df1['Emp_age']=df1['Emp_age'].fillna(0)
```

In [17]:
```
mean=df1['Emp_salary'].mean()
df1['Emp_salary'].fillna(mean,inplace=True)
```

In [18]:
```
mod=df1['Emp_place'].mode()

df1['Emp_place'].fillna(mod[0],inplace=True)
```

In [19]:
```
round(df1.isnull().mean()*100,2)
```

Out[19]:
```
Comp_name      0.0
Emp_age        0.0
Emp_salary     0.0
Emp_place      0.0
Emp_country    0.0
Emp_gender     0.0
dtype: float64
```

In [20]:
```
df1.shape[0]
```

Out[20]:
```
136
```

Outliers

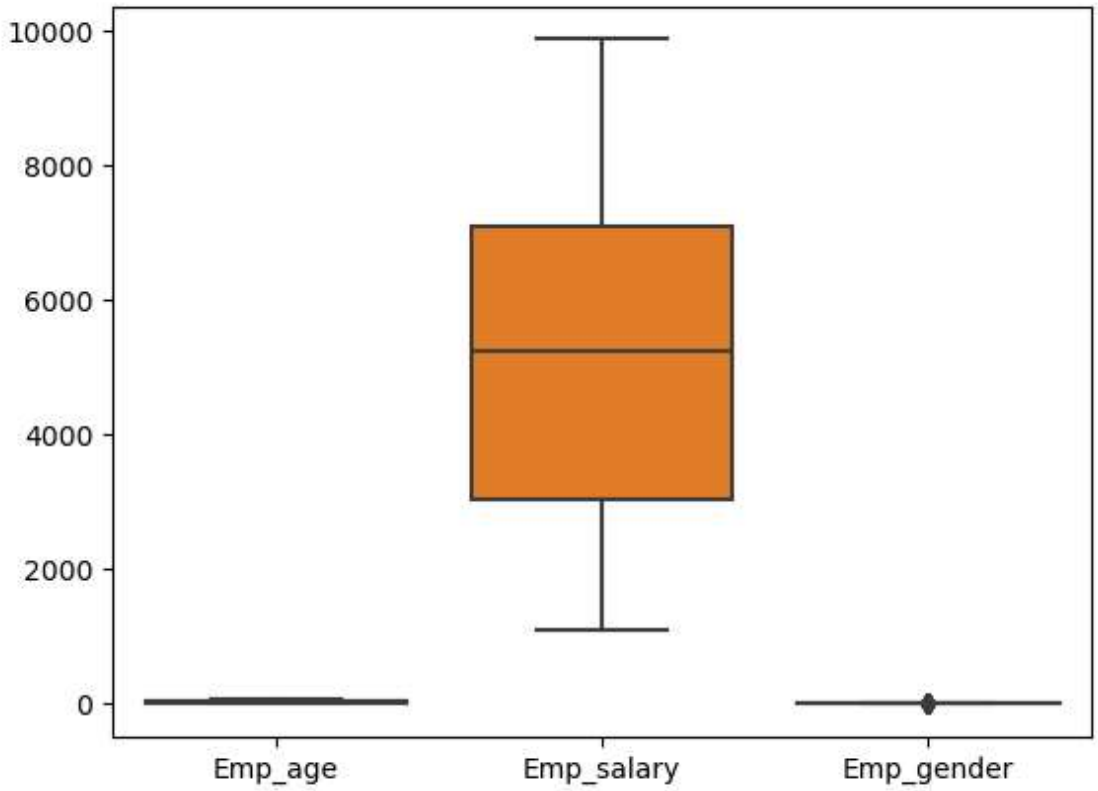
In [21]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [22]:
```
num=df1.select_dtypes('number')
num.skew()
```

Out[22]:
```
Emp_age      -0.643296
Emp_salary    0.214116
Emp_gender    1.311559
dtype: float64
```

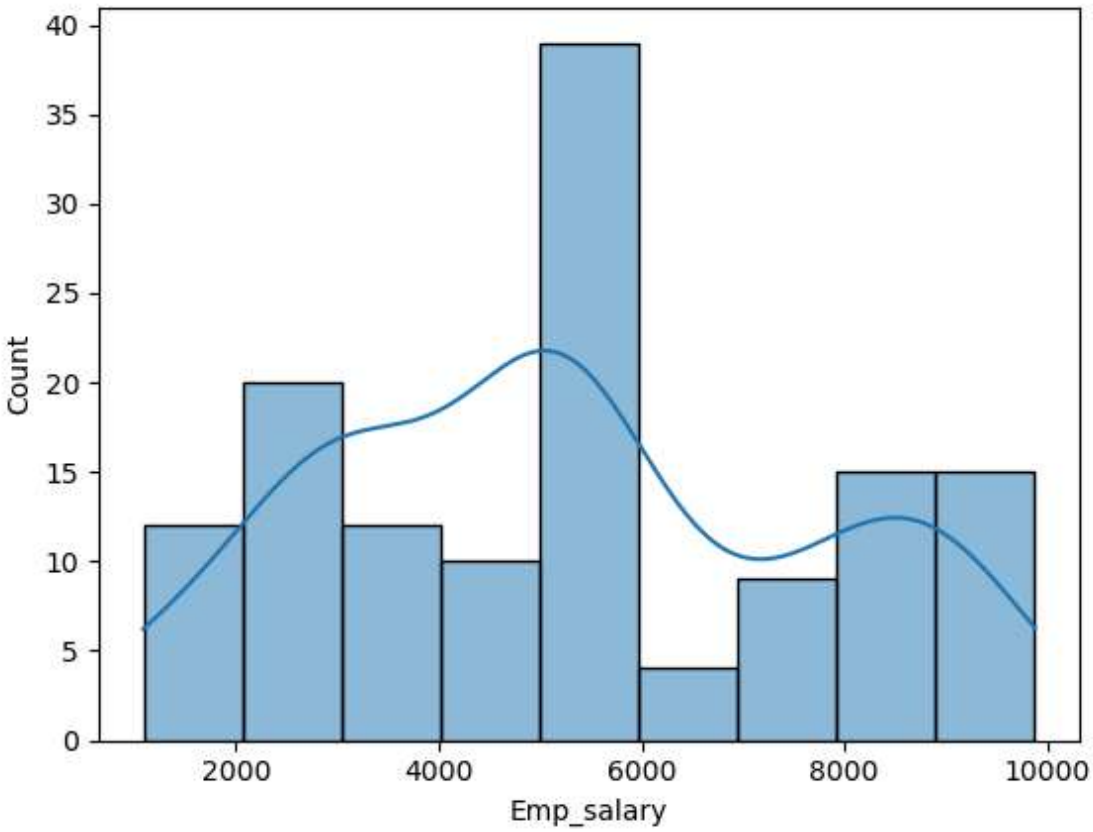
In [23]:
```
sns.boxplot(df1)
```

Out[23]: <Axes: >



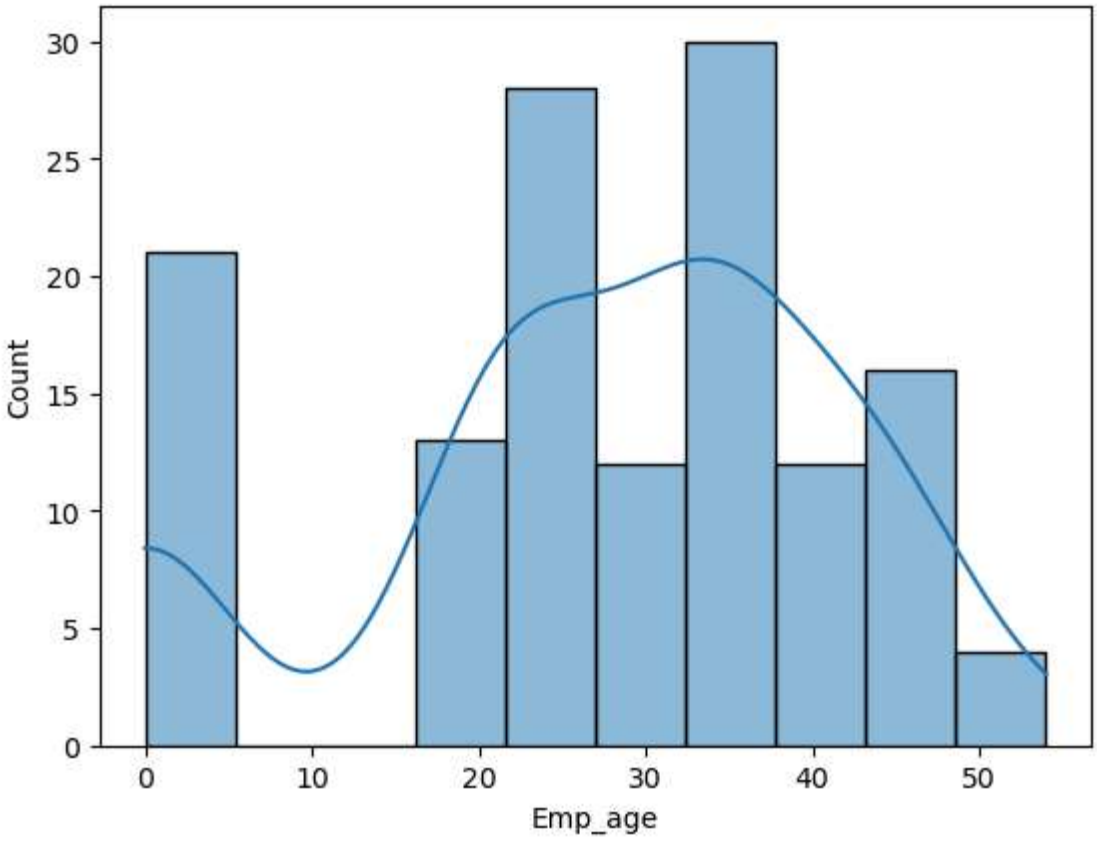
In [24]:
```
sns.histplot(df1['Emp_salary'],kde=True)
```

Out[24]: <Axes: xlabel='Emp_salary', ylabel='Count'>



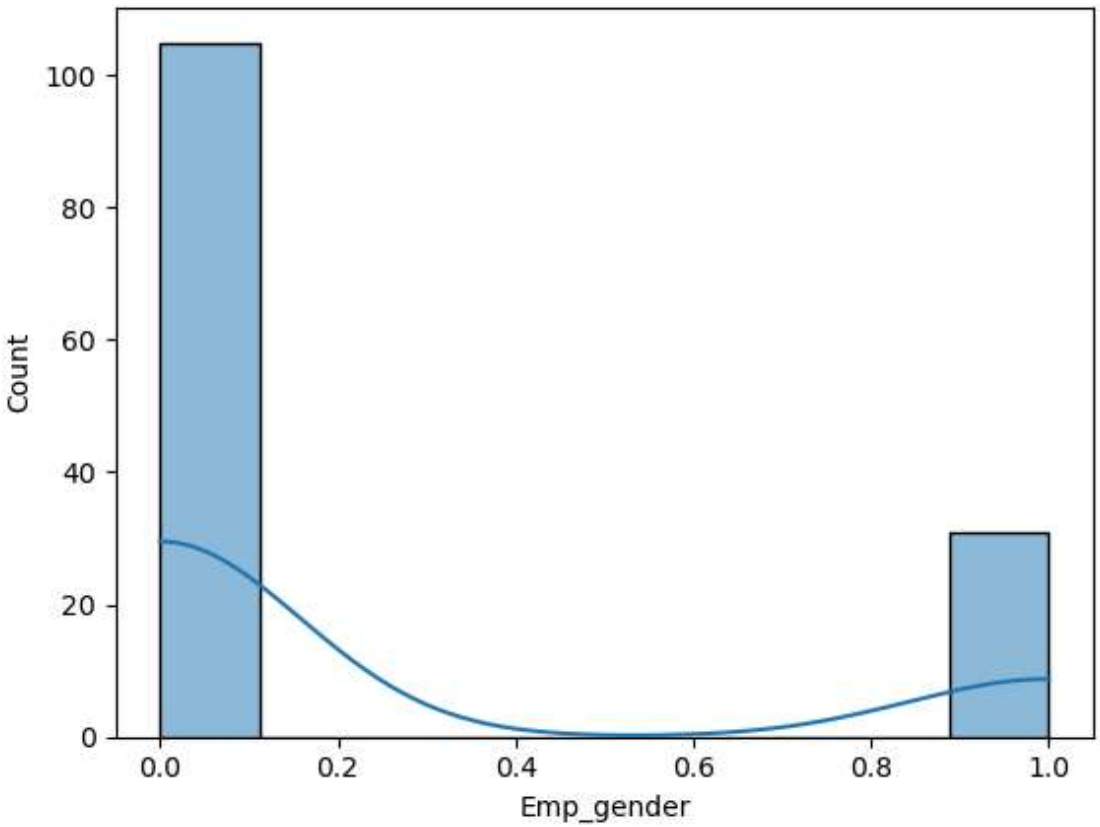
In [25]:
```
sns.histplot(df1['Emp_age'],kde=True)
```

Out[25]:
```
<Axes: xlabel='Emp_age', ylabel='Count'>
```



In [26]:
```
sns.histplot(df1['Emp_gender'],kde=True)
```

Out[26]: <Axes: xlabel='Emp_gender', ylabel='Count'>



Data Analysis:

Filter the data with age >40 and salary<5000

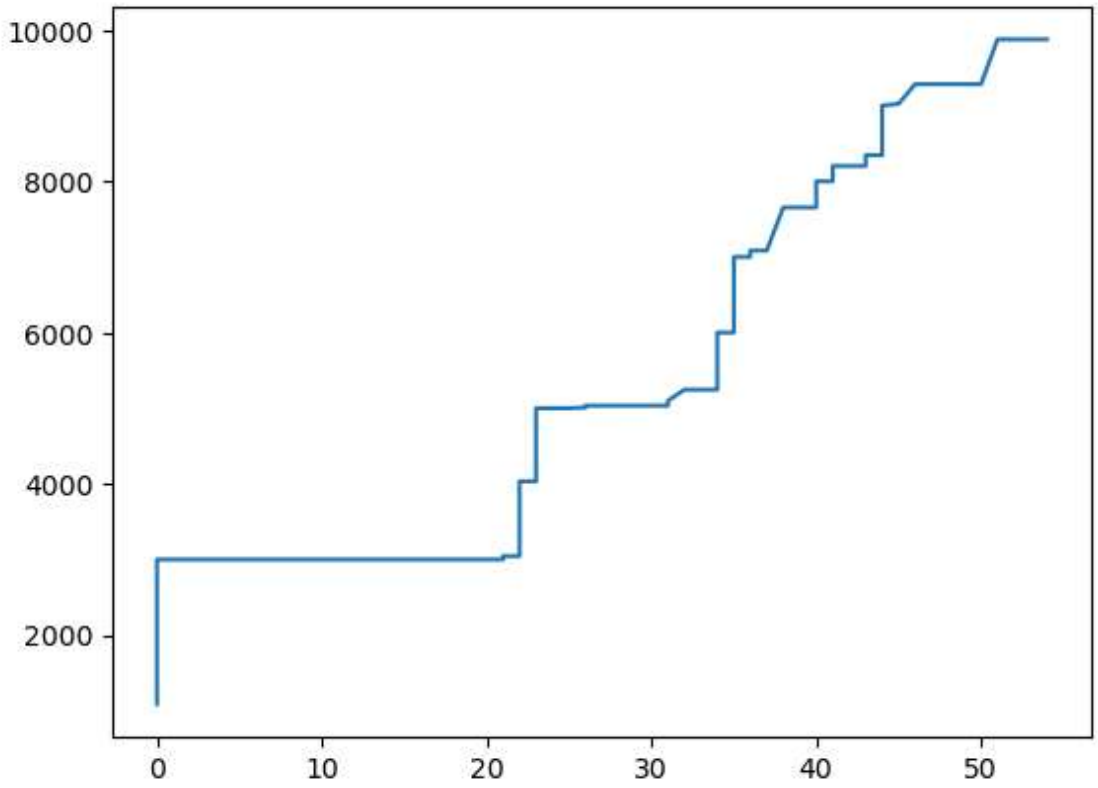
In [ ]:
```

```

In [27]:
```
filtred_data=df1[(df1['Emp_age']>40)&(df1['Emp_salary']<5000)]
filtred_data
```

Out[27]:

	Comp_name	Emp_age	Emp_salary	Emp_place	Emp_country	Emp_gender
21	Infosys	50.0	3184.0	Delhi	India	0
32	Infosys	45.0	4034.0	Calcutta	India	0
39	Infosys	41.0	3000.0	Mumbai	India	0
50	Infosys	41.0	3000.0	Chennai	India	0
57	Infosys	51.0	3184.0	Hyderabad	India	0
68	Infosys	43.0	4034.0	Mumbai	India	0
75	Infosys	44.0	3000.0	Cochin	India	0
86	Infosys	41.0	3000.0	Delhi	India	0
93	Infosys	54.0	3184.0	Mumbai	India	0
104	Infosys	44.0	4034.0	Delhi	India	0
122	Infosys	44.0	3234.0	Mumbai	India	0
129	Infosys	50.0	3184.0	Calcutta	India	0
138	CTS	44.0	3033.0	Cochin	India	0
140	Infosys	44.0	4034.0	Hyderabad	India	0
145	Infosys	44.0	4034.0	Delhi	India	1

In [28]:
```
x=df1['Emp_age'].sort_values(ascending=True)
y=df1['Emp_salary'].sort_values(ascending=True)
plt.plot(x,y)
plt.show()
```



In [29]:
```
df1['Emp_place'].value_counts()
```

Out[29]:
```
Emp_place
Mumbai        46
Calcutta      30
Chennai       13
Delhi         13
Cochin        13
Noida          7
Hyderabad      7
Podicherry     3
Pune           2
Bhopal         1
Nagpur         1
Name: count, dtype: int64
```

In [30]:
```
data=df1['Emp_place'].value_counts()
x=list(data.index)
y=df1['Emp_place'].value_counts().values
```

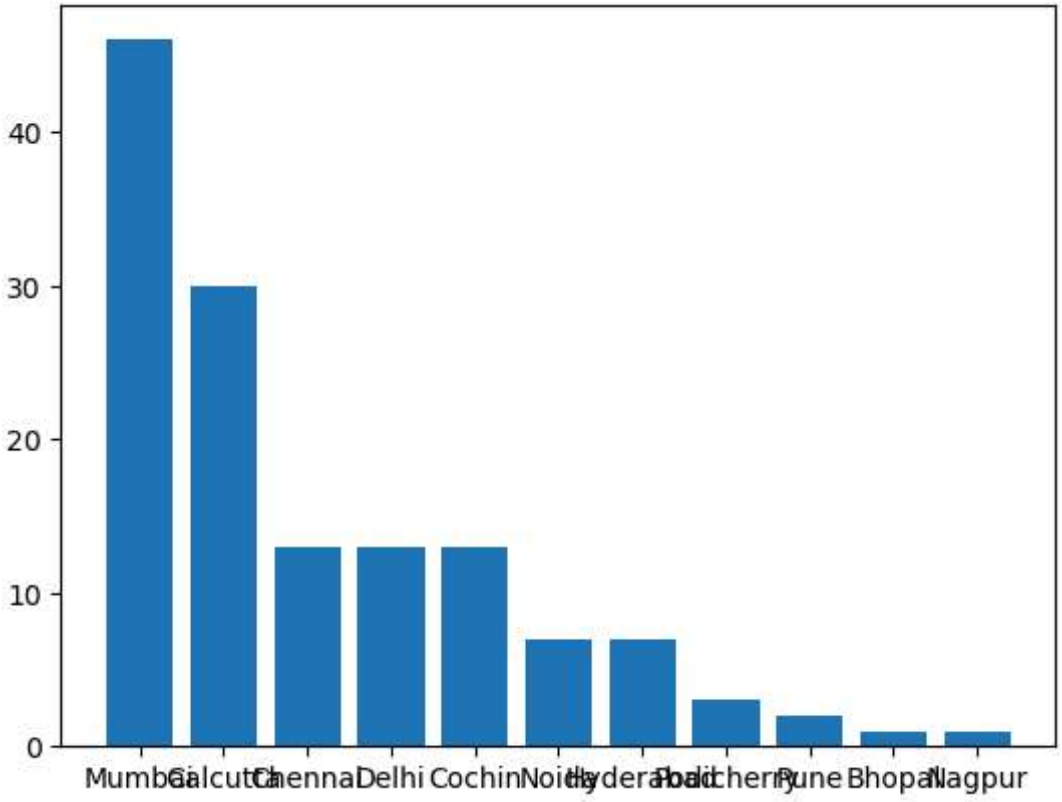
In [31]:
```
x
```

Out[31]:
```
['Mumbai',
 'Calcutta',
 'Chennai',
 'Delhi',
 'Cochin',
 'Noida',
 'Hyderabad',
 'Podicherry',
 'Pune',
 'Bhopal',
 'Nagpur']
```

In [32]:
```
y
```

Out[32]:
```
array([46, 30, 13, 13, 13,  7,  7,  3,  2,  1,  1], dtype=int64)
```

In [33]:
```
plt.bar(x,y)
plt.show()
```



Data Encoding:

In [34]:
```
df1
```

Out[34]:

	Comp_name	Emp_age	Emp_salary	Emp_place	Emp_country	Emp_gender
0	TCS	20.0	5244.974138	Chennai	India	0
1	Infosys	30.0	5244.974138	Mumbai	India	0
2	TCS	35.0	2300.000000	Calcutta	India	0
3	Infosys	40.0	3000.000000	Delhi	India	0
4	TCS	23.0	4000.000000	Mumbai	India	0
...	...	...	...	...	...	...
142	Infosys Pvt Lmt	22.0	8202.000000	Mumbai	India	0
143	TCS	33.0	9024.000000	Calcutta	India	1
145	Infosys	44.0	4034.000000	Delhi	India	1
146	TCS	33.0	5034.000000	Mumbai	India	1
147	Infosys	22.0	8202.000000	Cochin	India	0

136 rows × 6 columns

In [35]:
```
from sklearn import preprocessing
lbl_encoder=preprocessing.LabelEncoder()


df1['place_lbl_encoded']=lbl_encoder.fit_transform(df1['Emp_place'])
df1['Comp_name_lbl_encoded']=lbl_encoder.fit_transform(df1['Comp_name'])
df1
```

Out[35]:

	Comp_name	Emp_age	Emp_salary	Emp_place	Emp_country	Emp_gender	place_lbl
0	TCS	20.0	5244.974138	Chennai	India	0	
1	Infosys	30.0	5244.974138	Mumbai	India	0	
2	TCS	35.0	2300.000000	Calcutta	India	0	
3	Infosys	40.0	3000.000000	Delhi	India	0	
4	TCS	23.0	4000.000000	Mumbai	India	0	
...	...	...	...	...	...	...	
142	Infosys Pvt Lmt	22.0	8202.000000	Mumbai	India	0	
143	TCS	33.0	9024.000000	Calcutta	India	1	
145	Infosys	44.0	4034.000000	Delhi	India	1	
146	TCS	33.0	5034.000000	Mumbai	India	1	
147	Infosys	22.0	8202.000000	Cochin	India	0	

136 rows × 8 columns

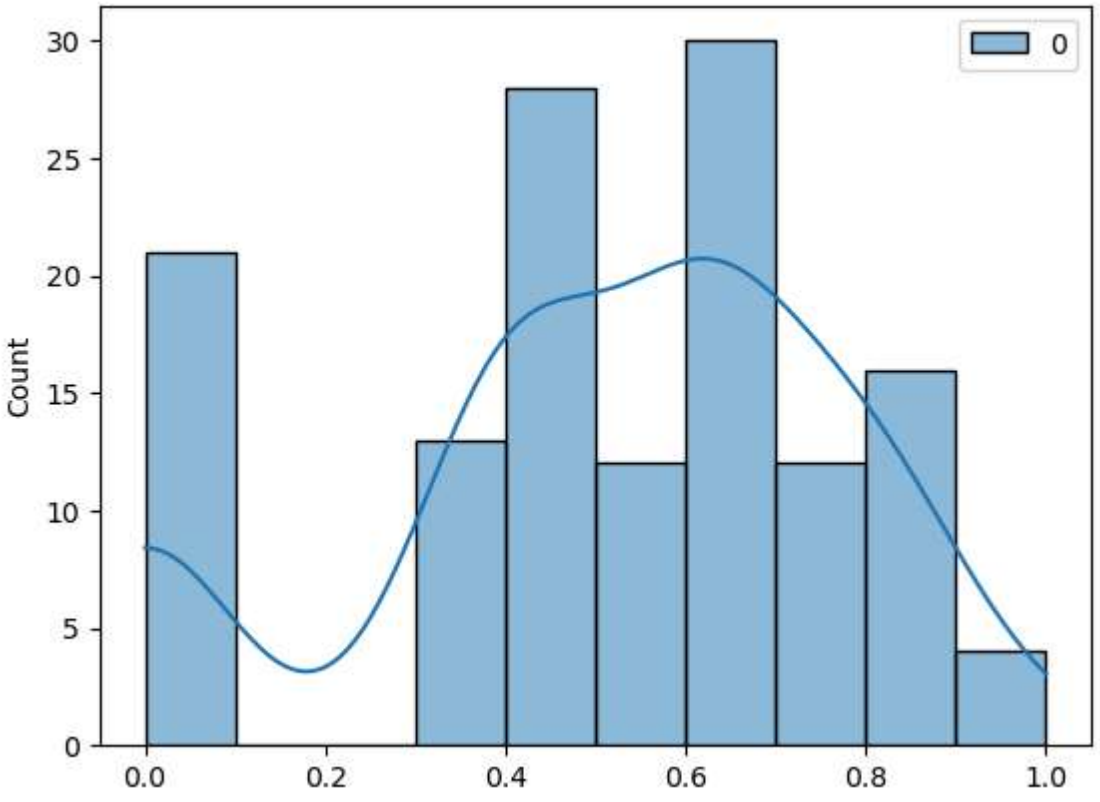
Feature Scaling:

In [36]:
```
from sklearn.preprocessing import MinMaxScaler,StandardScaler


min_max_scaler=MinMaxScaler()
x=df1[['Emp_age']]
age_min_max_scale= min_max_scaler.fit_transform(x)
age_min_max_scale
```

Out[36]:
```
array([[0.37037037],
       [0.55555556],
       [0.64814815],
       [0.74074074],
       [0.42592593],
       [0.        ],
       [0.        ],
       [0.42592593],
       [0.62962963],
       [0.83333333],
       [0.42592593],
       [0.62962963],
       [0.83333333],
       [0.33333333],
       [0.74074074],
       [0.42592593],
       [0.42592593],
       [0.62962963],
       [0.40740741],
```

In [37]:
```
sns.histplot(age_min_max_scale,kde=True)
```

Out[37]:
```
<Axes: ylabel='Count'>
```



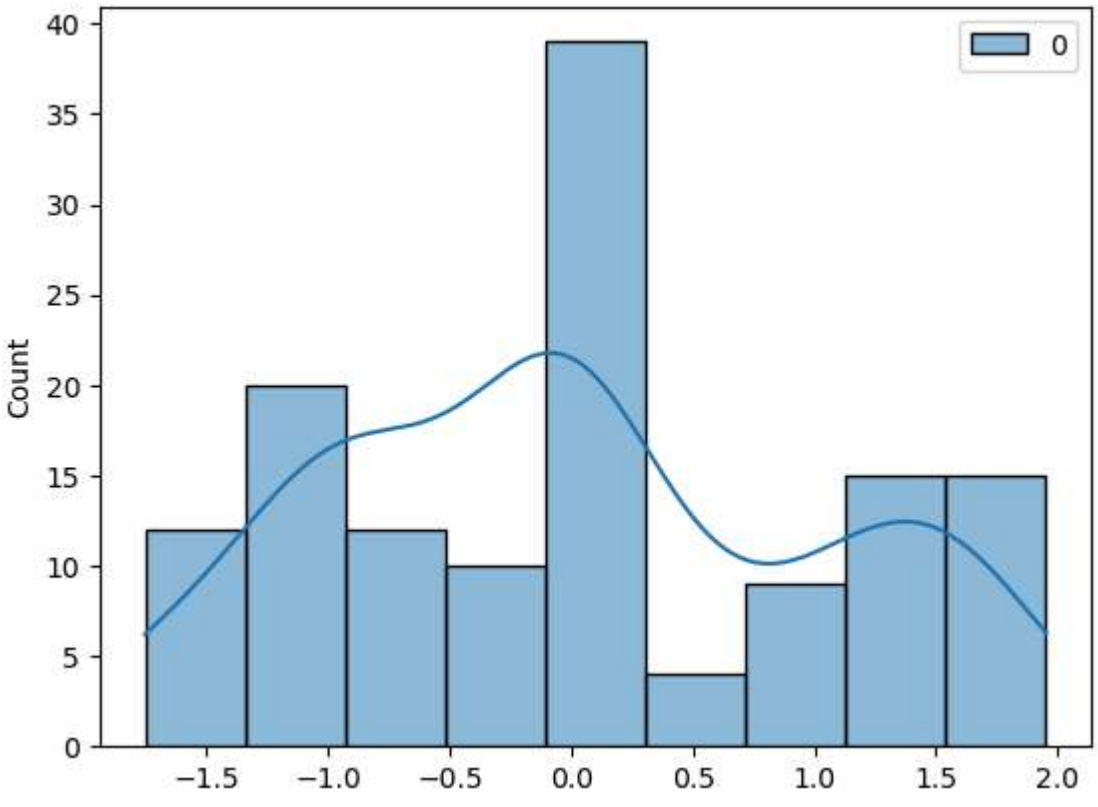
Standard scaler:

In [38]:
```
x=df1[['Emp_salary']]
standard_scaler=StandardScaler()
sal_standard_scale=standard_scaler.fit_transform(x)
sal_standard_scale
```

Out[38]:
```
array([[-3.83477899e-16],
       [-3.83477899e-16],
       [-1.24171421e+00],
       [-9.46567325e-01],
       [-5.24928915e-01],
       [-1.03290506e-01],
       [ 3.18347904e-01],
       [ 7.39986313e-01],
       [ 1.16162472e+00],
       [ 1.58326313e+00],
       [-3.83477899e-16],
       [-1.75231833e+00],
       [-3.83477899e-16],
       [-1.69118076e+00],
       [-9.46567325e-01],
       [-9.46567325e-01],
       [-9.33918173e-01],
       [-1.03290506e-01],
       [-3.83477899e-16],
```

In [39]:
```
sns.histplot(sal_standard_scale,kde=True)
```

Out[39]:
```
<Axes: ylabel='Count'>
```



SUMMERY

1. Importing and Exploring the Dataset

Load the dataset using *pd.read_csv()* and check the structure of the data using methods like *info(), head(), isnull().sum(), and describe()* to get a sense of missing values, basic statistics, and unique value counts in categorical columns.

Rename columns for better readability using *rename()* to give more meaningful names to your variables like Emp_age, Emp_salary, etc.

2. Data Cleaning

Remove duplicates: identify and remove duplicate rows using *duplicated().sum() and drop_duplicates()*. Handle missing values: Drop rows where Comp_name is missing. Fill missing values in the Emp_age column with 0. Fill missing values in Emp_salary with the column's mean. Fill missing values in Emp_place using the most frequent value (mode). Check the final shape of the dataset after cleaning to ensure the data has been processed correctly.

3. Outliers Detection

Box plots and histograms: Use seaborn and matplotlib to visualize the distribution of features like Emp_salary and Emp_age. Box plots help in detecting outliers, and histograms show the data distribution.

Skewness: compute skewness for numerical features to see how much the distribution deviates from normal.

4. Data Analysis

Filter Data: filter the dataset for employees whose age is greater than 40 and salary is less than 5000 using conditions in Pandas.

Visualizations: Plot the relationship between age and salary using *plt.plot()* to see any trends.

Bar chart: Show the distribution of employees across different places with a bar plot using *plt.bar()*.

5. Data Encoding

Label Encoding: apply label encoding on categorical columns like Emp_place and Comp_name to convert them into numerical values using *LabelEncoder() from sklearn.preprocessing*.

6. Feature Scaling

MinMax Scaling: scale the Emp_age column to fit within a range (0-1) using* MinMaxScaler()*.

Standard Scaling: scale the Emp_salary column to have zero mean and unit variance using *StandardScaler()*. The scaled data is then plotted using histograms to visualize the transformed distributions.

Final Output:

This dataset has been cleaned, encoded, and scaled, making it ready for further machine learning tasks. It have visualized important aspects of the data and handled key preprocessing steps like missing values, duplicates, outliers, and feature scaling.

In [ ]: