Practical No.2

1. Create Pandas DataFrame to enter employee database which includes Sr No, Name, Mobile No, City

Implementation:

- **Program:**

```
#pandas dataframe
#Employee
print("Employee_Details")
import pandas as pd
pd.DataFrame({"Sr No":[101,102,103,104,105],
              "Name":['Vaishnavi','Shubham','Shraddha','Rupesh','Nimish
'],
              "Mobile
No":[9011235123,9850564568,9645569841,9565231452,8896548654],
              "City":['Satara','Mumbai','Thane','Pune','Sangali']})
```

- **Output:**

```
Employee_Details
```

	Sr No	Name	Mobile No	City
0	101	Vaishnavi	9011235123	Satara
1	102	Shubham	9850564568	Mumbai
2	103	Shraddha	9645569841	Thane
3	104	Rupesh	9565231452	Pune
4	105	Nimish	8896548654	Sangali

2. Use Iris Dataset from Github and perform all basic operations on Iris Dataset

Implementation:

- **Program:**

```
url='https://gist.githubusercontent.com/curran/a08a1080b88344b0c8a7/raw
/0e7a9b0a5d22642a06d3d5b9bcbad9890c8ee534/iris.csv'
iris2=pd.read_csv(url)
iris2.head()
```

- **Output:**

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

- **Program:**

```
url='https://gist.githubusercontent.com/curran/a08a1080b88344b0c8a7/raw
/0e7a9b0a5d22642a06d3d5b9bcbad9890c8ee534/iris.csv'
iris3=pd.read_csv(url)
iris3.tail()
```

- **Output:**

	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

- **Program:**

```
iris2.shape
```

- **Output:**

```
(150, 5)
```

- **Program:**

```
iris2.describe()
```

- **Output:**

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

- **Program:**

```
iris1.iloc[5:10,2:]
```

- **Output:**

	1.4	0.2	Iris-setosa
5	1.4	0.3	Iris-setosa
6	1.5	0.2	Iris-setosa
7	1.4	0.2	Iris-setosa
8	1.5	0.1	Iris-setosa
9	1.5	0.2	Iris-setosa

- **Program:**

```
iris2.iloc[5:10,1:]
```

- **Output:**

	sepal_width	petal_length	petal_width	species
5	3.9	1.7	0.4	setosa
6	3.4	1.4	0.3	setosa
7	3.4	1.5	0.2	setosa
8	2.9	1.4	0.2	setosa
9	3.1	1.5	0.1	setosa

- **Program:**

```
iris2.iloc[5:9,3:]
```

- **Output:**

	petal_width	species
5	0.4	setosa
6	0.3	setosa
7	0.2	setosa
8	0.2	setosa

- **Program:**

```
iris2.loc[0:4,"sepal_width"]
```

- **Output**:

```
0    3.5
1    3.0
2    3.2
3    3.1
4    3.6
Name: sepal_width, dtype: float64
```

- **Program:**

```
iris2.loc[0:3,"petal_width"]
```

- **Output:**

```
0    0.2
1    0.2
2    0.2
3    0.2
Name: petal_width, dtype: float64
```

- **Program:**

```
iris2.drop('species',axis=1)
```

- **Output:**

	sepal_length	sepal_width	petal_length	petal_width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

- **Program:**

```
iris2['species'].value_counts()
```

- **Output:**

```
setosa        50
versicolor    50
virginica     50
Name: species, dtype: int64
```

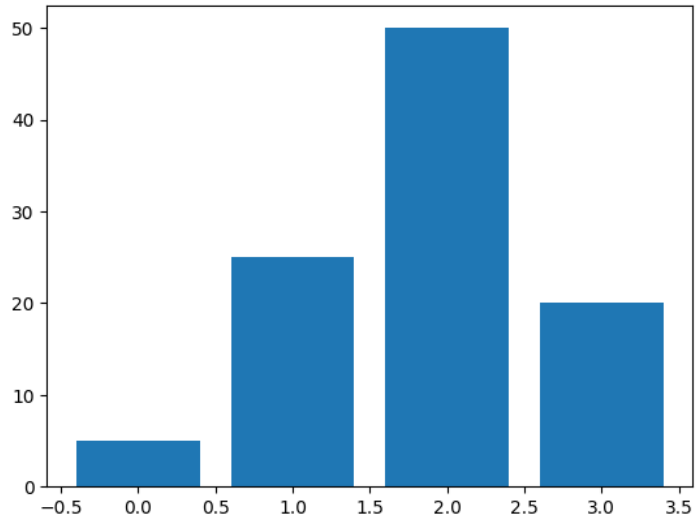
3. Use matplotlib to plot bar chart using dictionary

Implementation:

- **Program:**

```
import matplotlib.pyplot as plt
data=[5.,25.,50.,20.]
plt.bar(range(len(data)),data)
plt.show()
```

- **Output:**



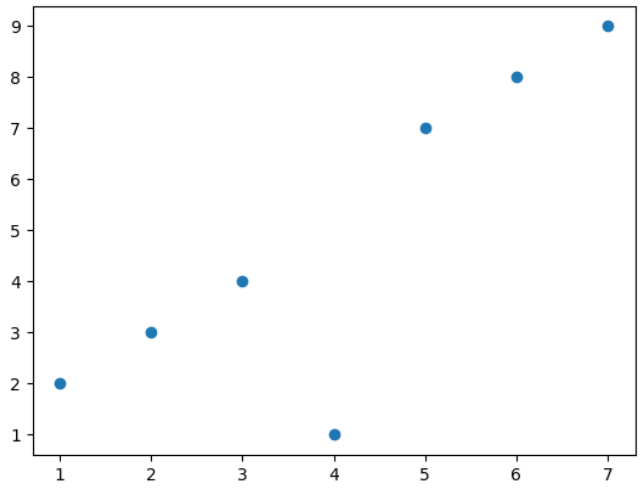
4. Use matplotlib to scatter graph and histogram

Implementation:

- **Program:**

```
x=[1,2,3,4,5,6,7]
y=[2,3,4,1,7,8,9]
plt.scatter(x,y)
plt.show()
```

- **Output:**



- **Program:**

```
import matplotlib.pyplot as plt
import numpy as np

x = np.random.normal(170, 10, 250)

plt.hist(x)
plt.show()
```

- **Output:**