In [1]:

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

In [2]:

```
df=pd.read_csv(r"C:\Users\Admin\Downloads\16_Sleep_health_and_lifestyle_dataset - 16_Sleep_health_and_lifestyle_dataset.
df
```

Out[2]:

	Person ID	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Blood Pressure	Heart Rate	Daily Steps	Sleep Disorder
0	1	Male	27	Software Engineer	6.1	6	42	6	Overweight	126/83	77	4200	None
1	2	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	None
2	3	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	None
3	4	Male	28	Sales Representative	5.9	4	30	8	Obese	140/90	85	3000	Sleep Apnea
4	5	Male	28	Sales Representative	5.9	4	30	8	Obese	140/90	85	3000	Sleep Apnea
...	...	...	...	...	...	...	...	...	...	...	...	...	...
369	370	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68	7000	Sleep Apnea
370	371	Female	59	Nurse	8.0	9	75	3	Overweight	140/95	68	7000	Sleep Apnea
371	372	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68	7000	Sleep Apnea
372	373	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68	7000	Sleep Apnea
373	374	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68	7000	Sleep Apnea

374 rows × 13 columns

In [3]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 374 entries, 0 to 373
Data columns (total 13 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Person ID                374 non-null    int64  
 1   Gender                   374 non-null    object 
 2   Age                      374 non-null    int64  
 3   Occupation               374 non-null    object 
 4   Sleep Duration           374 non-null    float64
 5   Quality of Sleep         374 non-null    int64  
 6   Physical Activity Level  374 non-null    int64  
 7   Stress Level             374 non-null    int64  
 8   BMI Category             374 non-null    object 
 9   Blood Pressure           374 non-null    object 
 10  Heart Rate               374 non-null    int64  
 11  Daily Steps              374 non-null    int64  
 12  Sleep Disorder           374 non-null    object 
dtypes: float64(1), int64(7), object(5)
memory usage: 38.1+ KB
```

In [4]:

```
df.describe()
```

Out[4]:

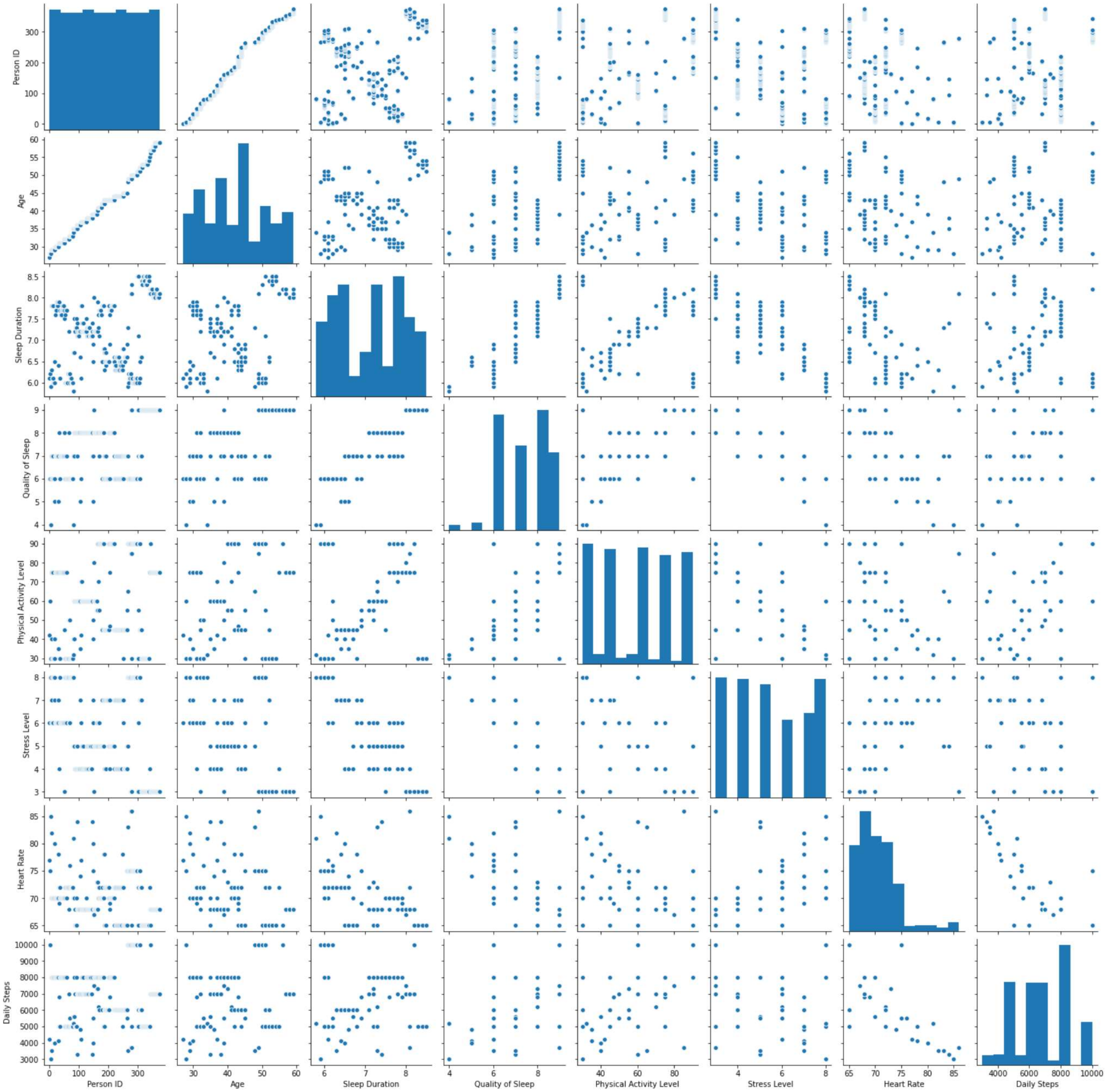
	Person ID	Age	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	Heart Rate	Daily Steps
count	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000
mean	187.500000	42.184492	7.132086	7.312834	59.171123	5.385027	70.165775	6816.844920
std	108.108742	8.673133	0.795657	1.196956	20.830804	1.774526	4.135676	1617.915679
min	1.000000	27.000000	5.800000	4.000000	30.000000	3.000000	65.000000	3000.000000
25%	94.250000	35.250000	6.400000	6.000000	45.000000	4.000000	68.000000	5600.000000
50%	187.500000	43.000000	7.200000	7.000000	60.000000	5.000000	70.000000	7000.000000
75%	280.750000	50.000000	7.800000	8.000000	75.000000	7.000000	72.000000	8000.000000
max	374.000000	59.000000	8.500000	9.000000	90.000000	8.000000	86.000000	10000.000000

```
In [5]: df.columns
```

```
Out[5]: Index(['Person ID', 'Gender', 'Age', 'Occupation', 'Sleep Duration',
               'Quality of Sleep', 'Physical Activity Level', 'Stress Level',
               'BMI Category', 'Blood Pressure', 'Heart Rate', 'Daily Steps',
               'Sleep Disorder'],
              dtype='object')
```

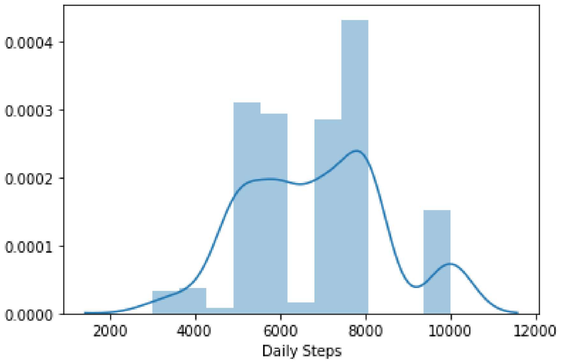
```
In [6]: sns.pairplot(df)
```

```
Out[6]: <seaborn.axisgrid.PairGrid at 0x16187f451c0>
```



In [7]:
```
sns.distplot(df['Daily Steps'])
```

Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x161894d0700>
```



In [8]:
```
df1=df[[ 'Age','Sleep Duration','Quality of Sleep', 'Physical Activity Level', 'Stress Level','Heart Rate', 'Daily Steps
df1
```

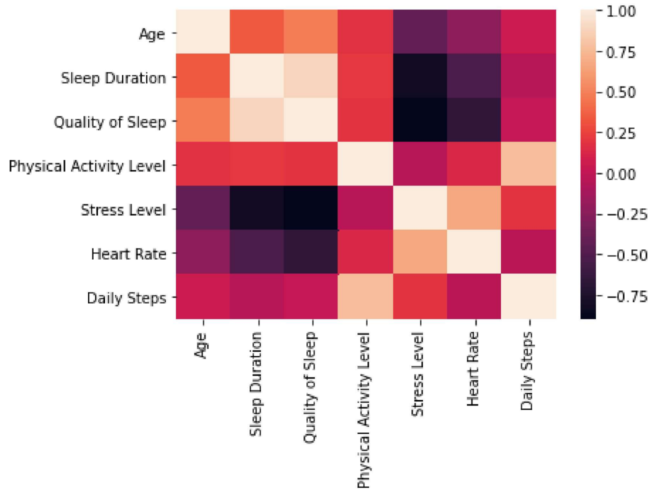
Out[8]:

	Age	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	Heart Rate	Daily Steps
0	27	6.1	6	42	6	77	4200
1	28	6.2	6	60	8	75	10000
2	28	6.2	6	60	8	75	10000
3	28	5.9	4	30	8	85	3000
4	28	5.9	4	30	8	85	3000
...	...	...	...	...	...	...	...
369	59	8.1	9	75	3	68	7000
370	59	8.0	9	75	3	68	7000
371	59	8.1	9	75	3	68	7000
372	59	8.1	9	75	3	68	7000
373	59	8.1	9	75	3	68	7000

374 rows × 7 columns

In [9]:
```
sns.heatmap(df1.corr())
```

Out[9]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1618b4ee1f0>
```



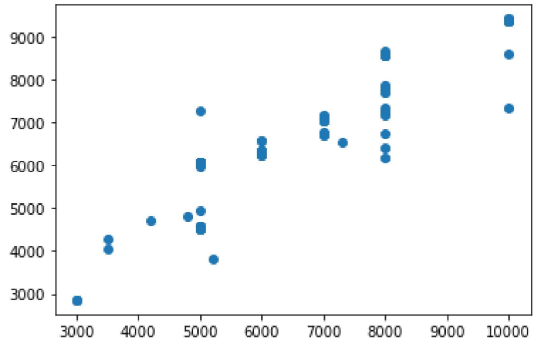
In [10]:
```
x=df1[['Age','Sleep Duration','Quality of Sleep', 'Physical Activity Level', 'Stress Level','Heart Rate']]
y=df[['Daily Steps']]
```

In [11]:
```
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)
```

```
In [12]: from sklearn.linear_model import LinearRegression
         lr= LinearRegression()
         lr.fit(x_train,y_train)

Out[12]: LinearRegression()
```

```
In [13]: print(lr.intercept_)

         [14488.78481287]
```

```
In [14]: prediction= lr.predict(x_test)
         plt.scatter(y_test,prediction)

Out[14]: <matplotlib.collections.PathCollection at 0x1618bfd91f0>
```



```
In [15]: print(lr.score(x_test,y_test))

         0.8260503686207115
```

```
In [16]: print(lr.score(x_train,y_train))

         0.7913066187858898
```

```
In [17]: from sklearn.linear_model import Ridge,Lasso
```

```
In [18]: rr=Ridge(alpha=10)
         rr.fit(x_train,y_train)

Out[18]: Ridge(alpha=10)
```

```
In [19]: rr.score(x_test,y_test)

Out[19]: 0.8260302396054762
```

```
In [20]: la=Lasso(alpha=10)
         la.fit(x_train,y_train)

Out[20]: Lasso(alpha=10)
```

```
In [21]: la.score(x_test,y_test)

Out[21]: 0.8260864769394787
```

```
In [ ]:
```