In [23]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [24]:

```
data = pd.read_csv("healthy_lifestyle_city.csv")
```

In [26]:

```
data.head()
```

Out[26]:

	City	Rank	Sunshine hours(City)	Cost of a bottle of water(City)	Obesity levels(Country)	Life expectancy(years) (Country)	Pollution(Index score) (City)
0	Amsterdam	1	1858	£1.92	20.40%	81.2	30.93
1	Sydney	2	2636	£1.48	29.00%	82.1	26.86
2	Vienna	3	1884	£1.94	20.10%	81.0	17.33
3	Stockholm	4	1821	£1.72	20.60%	81.8	19.63
4	Copenhagen	5	1630	£2.19	19.70%	79.8	21.24

In [27]:

```
data.tail()
```

Out[27]:

	City	Rank	Sunshine hours(City)	Cost of a bottle of water(City)	Obesity levels(Country)	Life expectancy(years) (Country)	Pollution(Index score) (City)
39	Milan	40	1915	£1.15	19.90%	82.7	67.19
40	Washington, D.C.	41	2528	£1.45	36.20%	78.8	39.18
41	New York	42	2535	£1.32	36.20%	78.8	57.36
42	Moscow	43	1901	£0.41	23.10%	69.5	57.63
43	Mexico City	44	2555	£0.45	28.90%	76.4	82.78

In [28]:

```
data.shape
```

Out[28]:

```
(44, 12)
```

In [29]:

```
data.columns
```

Out[29]:

```
Index(['City', 'Rank', 'Sunshine hours(City)',
       'Cost of a bottle of water(City)', 'Obesity levels(Country)',
       'Life expectancy(years) (Country)', 'Pollution(Index score) (Cit
y)',
       'Annual avg. hours worked', 'Happiness levels(Country)',
       'Outdoor activities(City)', 'Number of take out places(City)',
       'Cost of a monthly gym membership(City)'],
      dtype='object')
```

In [30]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 44 entries, 0 to 43
Data columns (total 12 columns):
 #   Column                                  Non-Null Count  Dtype  
---  ------                                  --------------  -----  
 0   City                                    44 non-null     object 
 1   Rank                                    44 non-null     int64  
 2   Sunshine hours(City)                    44 non-null     object 
 3   Cost of a bottle of water(City)         44 non-null     object 
 4   Obesity levels(Country)                 44 non-null     object 
 5   Life expectancy(years) (Country)        44 non-null     float64
 6   Pollution(Index score) (City)           44 non-null     object 
 7   Annual avg. hours worked                44 non-null     object 
 8   Happiness levels(Country)               44 non-null     float64
 9   Outdoor activities(City)                44 non-null     int64  
 10  Number of take out places(City)         44 non-null     int64  
 11  Cost of a monthly gym membership(City)  44 non-null     object 
dtypes: float64(2), int64(3), object(7)
memory usage: 4.2+ KB
```

In [31]:

```
data.describe()
```

Out[31]:

	Rank	Life expectancy(years) (Country)	Happiness levels(Country)	Outdoor activities(City)	Number of take out places(City)
count	44.000000	44.00000	44.000000	44.000000	44.000000
mean	22.500000	78.17500	6.435000	213.977273	1443.113636
std	12.845233	5.30437	0.991202	127.190297	1388.803270
min	1.000000	56.30000	3.570000	23.000000	250.000000
25%	11.750000	75.40000	5.870000	125.250000	548.000000
50%	22.500000	80.40000	6.900000	189.500000	998.000000
75%	33.250000	81.80000	7.175000	288.250000	1674.250000
max	44.000000	83.20000	7.800000	585.000000	6417.000000

In [32]:

```
data.isnull().sum()
```

Out[32]:

```
City                                  0
Rank                                  0
Sunshine hours(City)                  0
Cost of a bottle of water(City)       0
Obesity levels(Country)               0
Life expectancy(years) (Country)      0
Pollution(Index score) (City)         0
Annual avg. hours worked              0
Happiness levels(Country)             0
Outdoor activities(City)              0
Number of take out places(City)       0
Cost of a monthly gym membership(City)    0
dtype: int64
```

In [33]:

```
data1 = data.groupby(['City']).mean()
data1.sort_values(by = 'Rank', ascending = True)
```

Out[33]:

	Rank	Life expectancy(years) (Country)	Happiness levels(Country)	Outdoor activities(City)	Number of take out places(City)
City					
Amsterdam	1	81.2	7.44	422	1048
Sydney	2	82.1	7.22	406	1103
Vienna	3	81.0	7.29	132	1008
Stockholm	4	81.8	7.35	129	598
Copenhagen	5	79.8	7.64	154	523
Helsinki	6	80.4	7.80	113	309
Fukuoka	7	83.2	5.87	35	539
Berlin	8	80.6	7.07	254	1729
Barcelona	9	82.2	6.40	585	2344
Vancouver	10	81.7	7.23	218	788
Melbourne	11	82.1	7.22	243	813
Beijing	12	75.4	5.12	223	261
Bangkok	13	74.1	5.99	377	1796
Buenos Aires	14	75.9	5.97	246	1435
Toronto	15	81.7	7.23	174	1656
Madrid	16	82.2	6.40	216	2491
Jakarta	17	68.5	5.28	114	833
Seoul	18	81.3	5.87	144	389
Frankfurt	19	80.6	7.07	23	551
Geneva	20	82.6	7.56	44	444
Tel Aviv	21	81.9	7.12	139	420
Istanbul	22	74.7	5.13	419	934
Cairo	23	70.7	4.15	323	250
Taipei	24	75.4	5.12	134	717
Los Angeles	25	78.8	6.94	223	1439
Mumbai	26	67.3	3.57	187	1183
Boston	27	78.8	6.94	88	588
Dublin	28	80.5	7.09	159	659
Tokyo	29	83.2	5.87	387	5802
Chicago	30	78.8	6.94	171	1320

	Rank	Life expectancy(years) (Country)	Happiness levels(Country)	Outdoor activities(City)	Number of take out places(City)
City					
Hong Kong	31	75.4	5.51	277	1257
Shanghai	32	75.4	5.12	108	346
Brussels	33	80.4	6.86	55	988
San Francisco	34	78.8	6.94	242	1031
Paris	35	81.8	6.66	331	4363
Sao Paulo	36	73.9	6.37	158	3355
Zurich	37	82.6	7.56	69	538
London	38	80.4	7.16	433	6417
Johannesburg	39	56.3	4.81	194	492
Milan	40	82.7	6.38	110	2396
Washington, D.C.	41	78.8	6.94	83	744
New York	42	78.8	6.94	359	3081
Moscow	43	69.5	5.54	322	3206
Mexico City	44	76.4	6.46	192	1313

In [34]:

```
data1.sort_values(by = 'Rank', ascending = True).head()
```

Out[34]:

	Rank	Life expectancy(years) (Country)	Happiness levels(Country)	Outdoor activities(City)	Number of take out places(City)
City					
Amsterdam	1	81.2	7.44	422	1048
Sydney	2	82.1	7.22	406	1103
Vienna	3	81.0	7.29	132	1008
Stockholm	4	81.8	7.35	129	598
Copenhagen	5	79.8	7.64	154	523

In [35]:

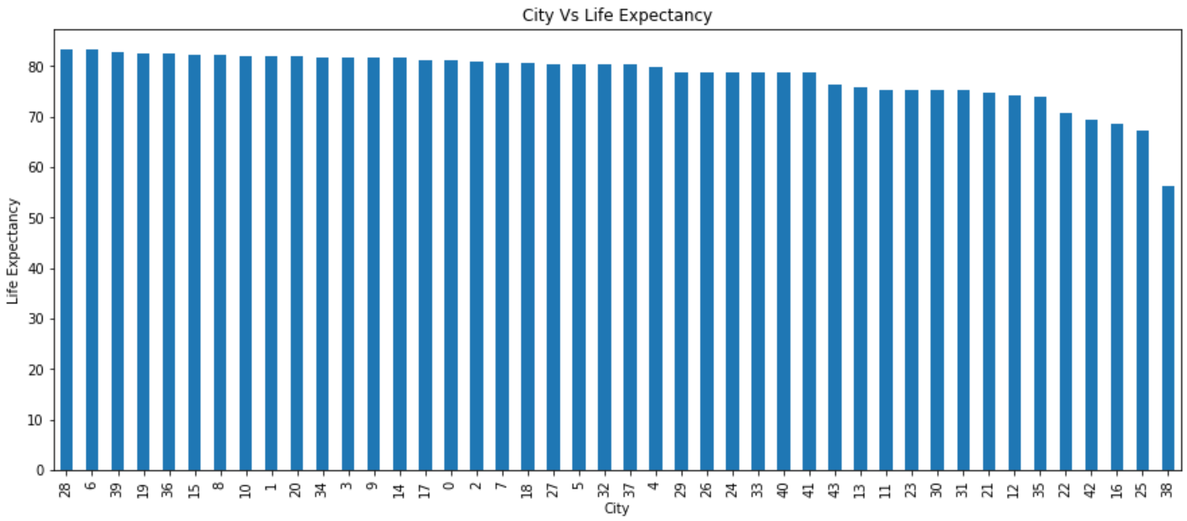
```
data1.sort_values(by = 'Rank', ascending = True).tail()
```

Out[35]:

City	Rank	Life expectancy(years) (Country)	Happiness levels(Country)	Outdoor activities(City)	Number of take out places(City)
Milan	40	82.7	6.38	110	2396
Washington, D.C.	41	78.8	6.94	83	744
New York	42	78.8	6.94	359	3081
Moscow	43	69.5	5.54	322	3206
Mexico City	44	76.4	6.46	192	1313

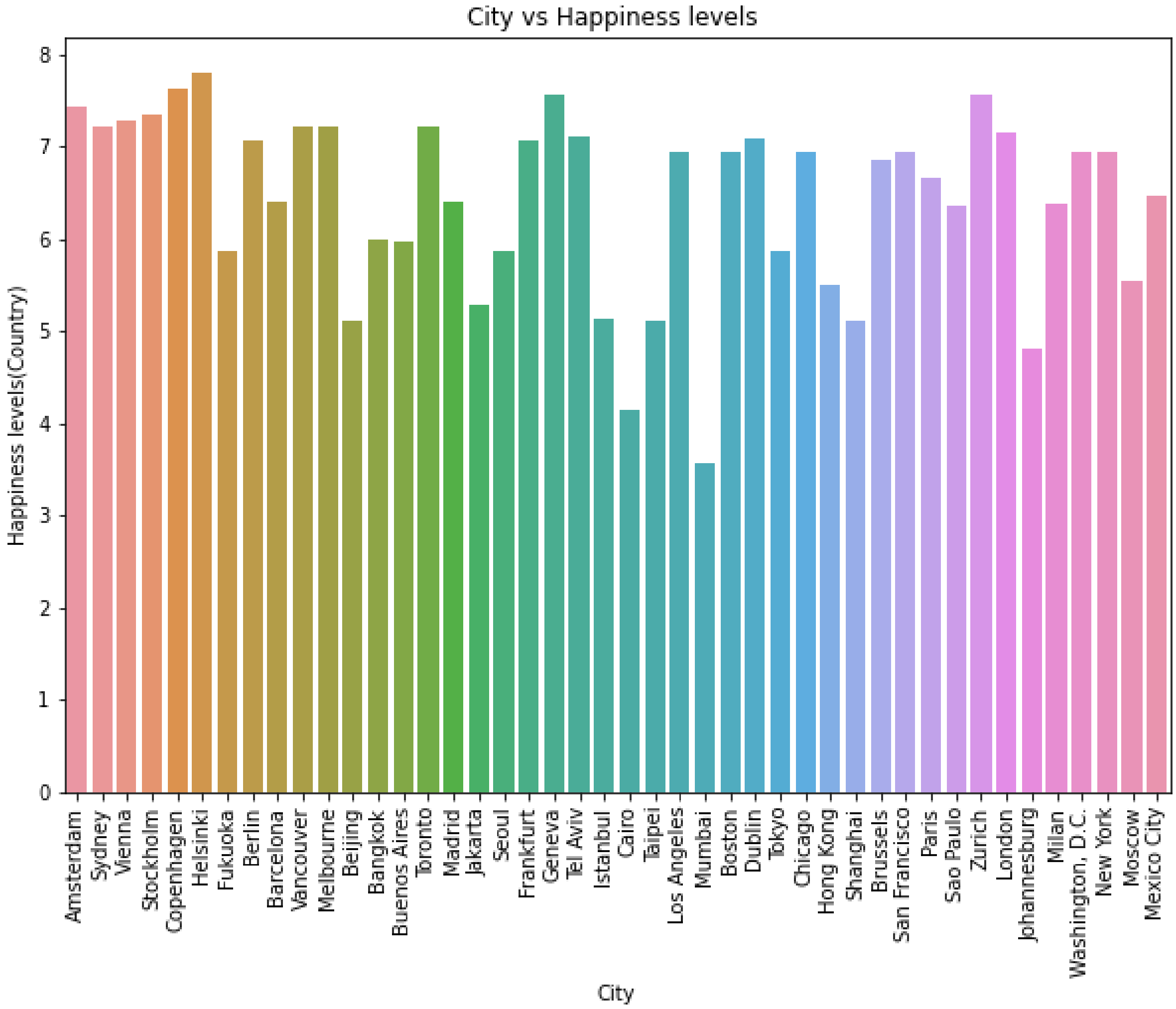
In [40]:

```
plt.subplots(figsize = (15, 6))
cr = data['Life expectancy(years) (Country)'].sort_values(ascending = False)
ax = cr.plot.bar()
ax.set_xlabel('City')
ax.set_ylabel('Life Expectancy')
ax.set_title('City Vs Life Expectancy')
plt.show()
```



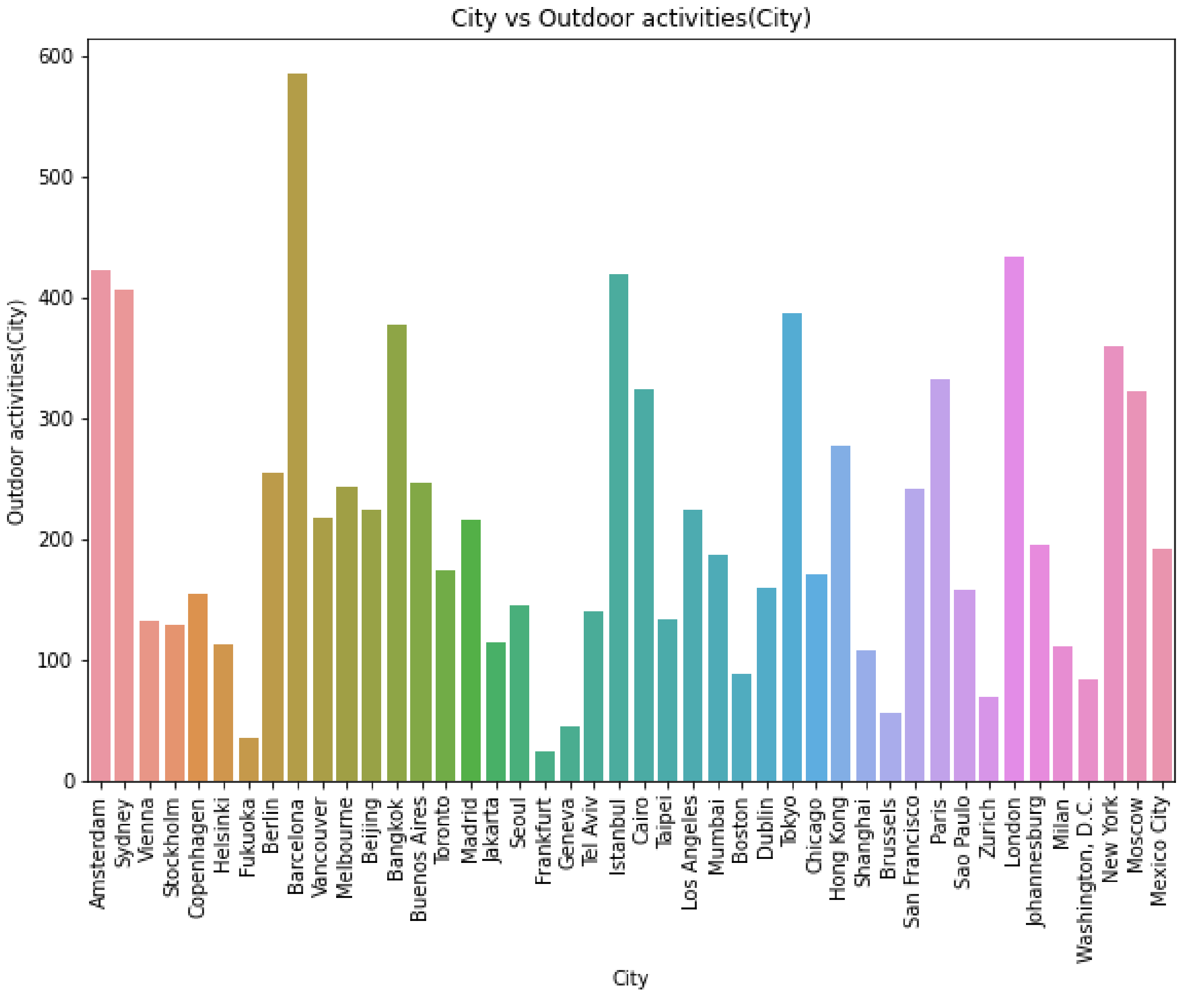
In [41]:

```
plt.figure(figsize = (10,7))
sns.barplot(data = data,x='City',y='Happiness levels(Country)')
plt.xticks(rotation=90)
plt.title("City vs Happiness levels ")
plt.show()
```



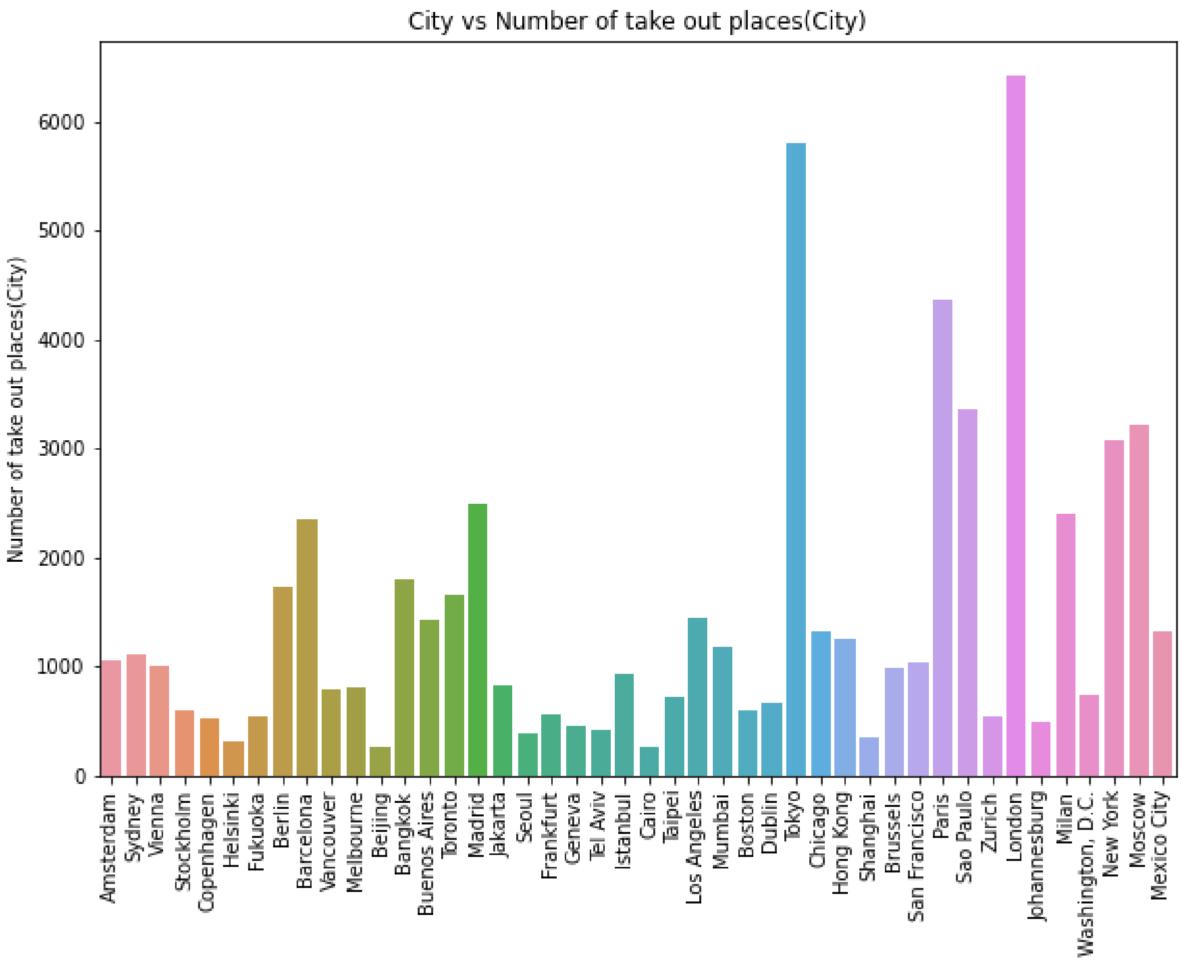
In [43]:

```
plt.figure(figsize = (10,7))
sns.barplot(data = data,x='City',y='Outdoor activities(City)')
plt.xticks(rotation=90)
plt.title("City vs Outdoor activities(City)")
plt.show()
```



In [44]:

```
plt.figure(figsize = (10,7))
sns.barplot(data = data,x='City',y='Number of take out places(City)')
plt.xticks(rotation=90)
plt.title("City vs Number of take out places(City)")
plt.show()
```



In [51]:

```
data2 = data.sort_values(by='Sunshine hours(City)',ascending=False)
data2.head()
```

Out[51]:

	City	Rank	Sunshine hours(City)	Cost of a bottle of water(City)	Obesity levels(Country)	Life expectancy(years) (Country)	Pollution(Inde score) (Cit
22	Cairo	23	3542	£0.16	32.00%	70.7	91.7
20	Tel Aviv	21	3311	£1.63	26.10%	81.9	47.2
24	Los Angeles	25	3254	£1.52	36.20%	78.8	66.0
38	Johannesburg	39	3124	£0.59	28.30%	56.3	61.8
33	San Francisco	34	3062	£1.60	36.20%	78.8	47.3

In [53]:

```
data['water bottle cost']=data['Cost of a bottle of water(City)'].str.slice(1,).astype(
data['obesity level']=data['Obesity levels(Country)'].str.slice(0,-1).astype(float)
data['gym cost']=data['Cost of a monthly gym membership(City)'].str.slice(1,).astype(fl
```

In [54]:

```
data['Sunshine hours(City)']=data['Sunshine hours(City)'].replace('-','0')
data['Pollution(Index score) (City)']=data['Pollution(Index score) (City)'].replace('-'
data['Annual avg. hours worked']=data['Annual avg. hours worked'].replace('-','0')
```

In [55]:

```
data=data.astype({'Sunshine hours(City)':float,'Pollution(Index score) (City)':float,'A
```

In [56]:

```
data_cleaned=data.drop(columns=['Cost of a bottle of water(City)','Obesity levels(Countr
```

In [57]:

```
data_cleaned.head()
```

Out[57]:

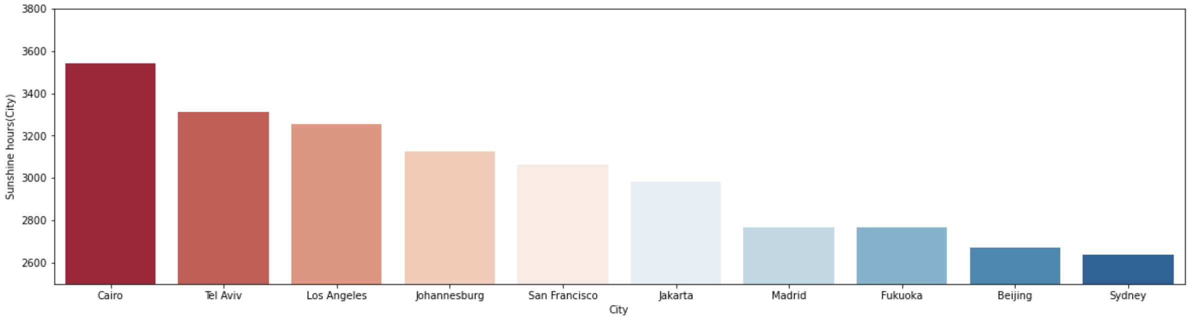
	City	Rank	Sunshine hours(City)	Life expectancy(years) (Country)	Pollution(Index score) (City)	Annual avg. hours worked	Happiness levels(Country)	ac
0	Amsterdam	1	1858.0	81.2	30.93	1434.0	7.44	
1	Sydney	2	2636.0	82.1	26.86	1712.0	7.22	
2	Vienna	3	1884.0	81.0	17.33	1501.0	7.29	
3	Stockholm	4	1821.0	81.8	19.63	1452.0	7.35	
4	Copenhagen	5	1630.0	79.8	21.24	1380.0	7.64	

In [58]:

```
data2=data_cleaned.sort_values(by='Sunshine hours(City)',ascending=False).head(10)
plt.figure(figsize=(20,5))
plt.ylim(2500,3800)
sns.barplot(data=data2, x='City',y='Sunshine hours(City)',palette='RdBu')
```

Out[58]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x618dcac2e0>
```

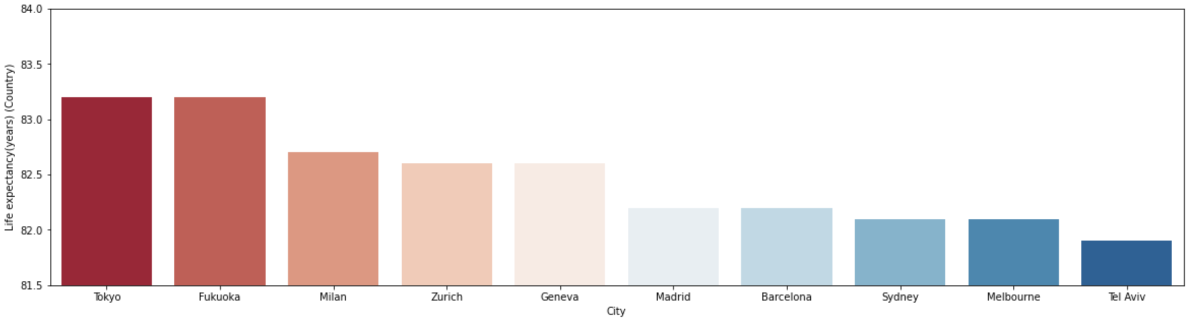


In [59]:

```
data3=data_cleaned.sort_values(by='Life expectancy(years) (Country)',ascending=False).he
plt.figure(figsize=(20,5))
plt.ylim(81.5,84)
sns.barplot(data=data3, x='City',y='Life expectancy(years) (Country)',palette='RdBu')
```

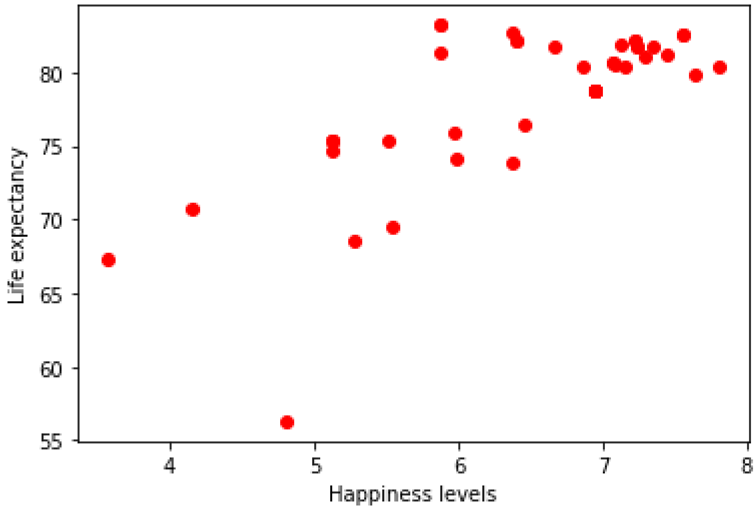
Out[59]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x61a1879790>
```



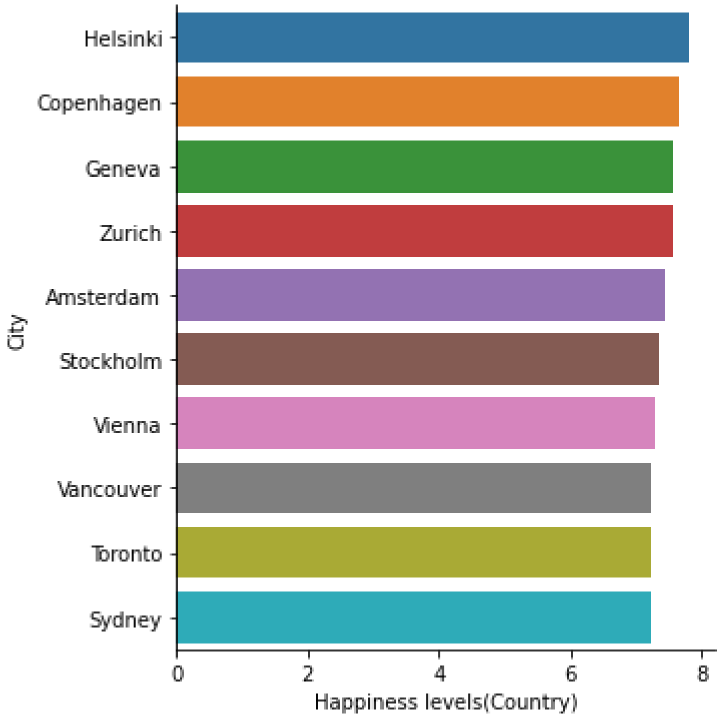
In [61]:

```
plt.scatter(data["Happiness levels(Country)"],data['Life expectancy(years) (Country)'],
plt.xlabel("Happiness levels");
plt.ylabel("Life expectancy");
```



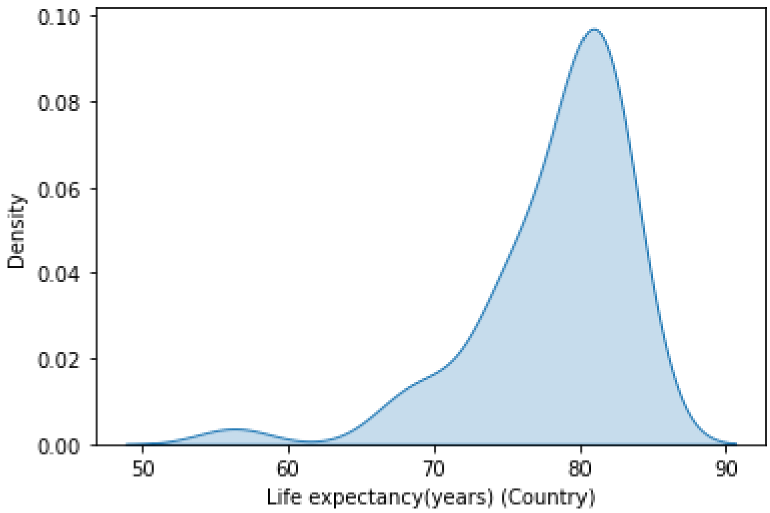
In [62]:

```
sns.catplot(x="Happiness levels(Country)", y="City", kind="bar", data=data.nlargest(10,
```



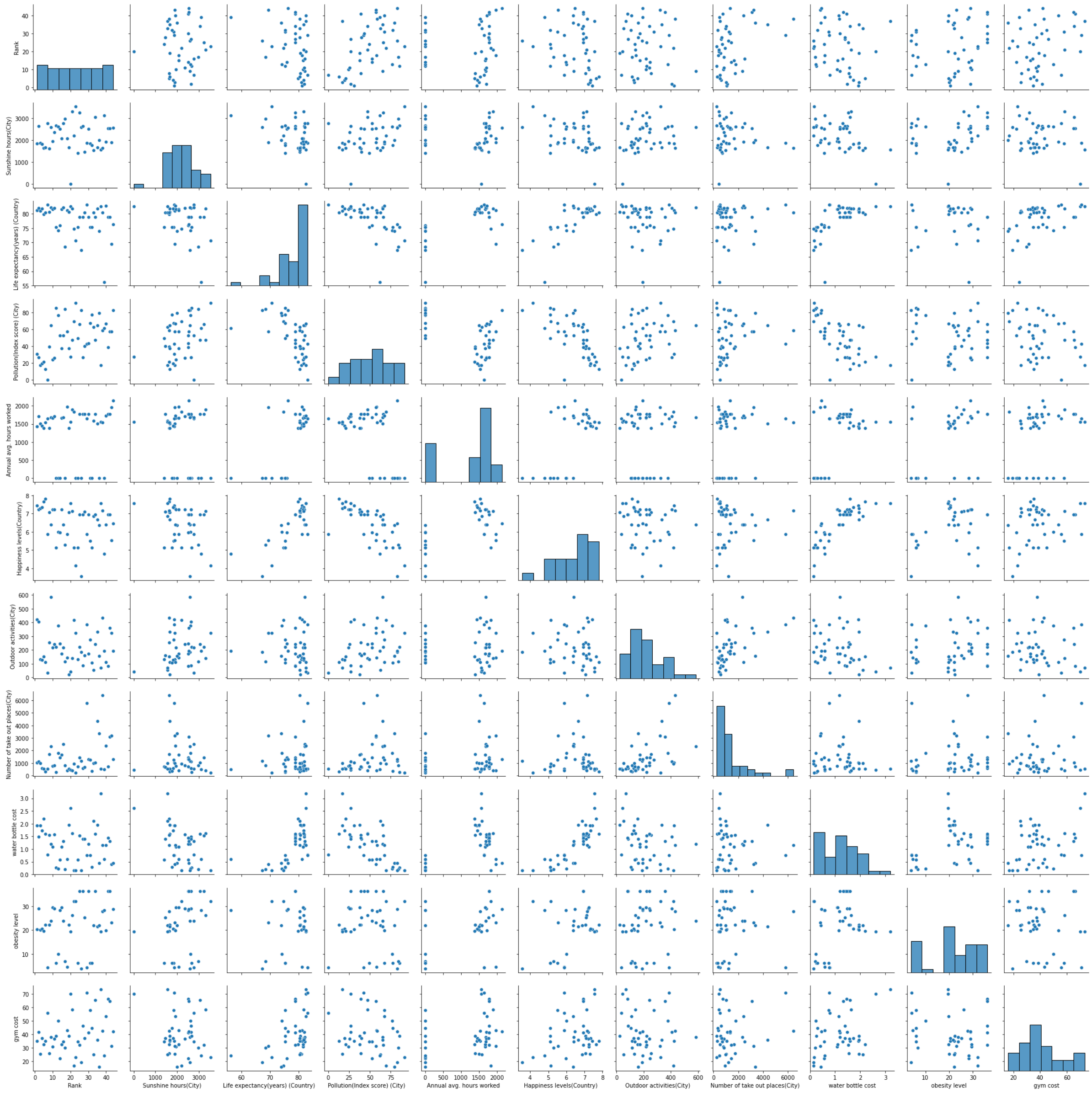
In [63]:

```
sns.kdeplot(x='Life expectancy(years) (Country)', data=data, shade=True);
```



In [64]:

```
sns.pairplot(data);
```

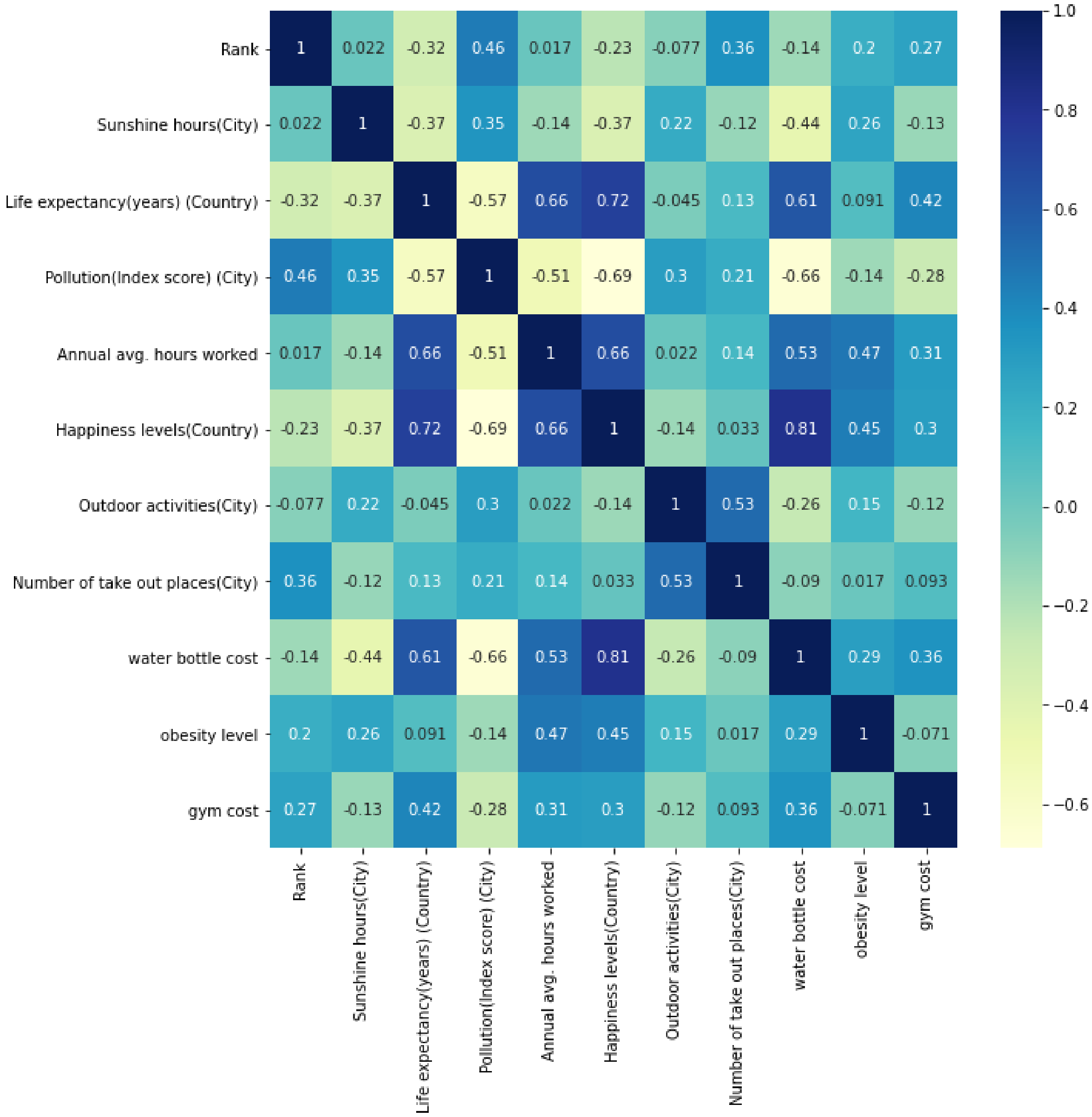


In [60]:

```
corr=data.corr()
plt.figure(figsize=(10,10))
sns.heatmap(corr,annot=True,cmap='YlGnBu')
```

Out[60]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x61a1832c40>
```



	Rank	Sunshine hours(City)	Life expectancy(years) (Country)	Pollution(Index score) (City)	Annual avg. hours worked	Happiness levels(Country)	Outdoor activities(City)	Number of take out places(City)	water bottle cost	obesity level	gym cost
Rank	1	0.022	-0.32	0.46	0.017	-0.23	-0.077	0.36	-0.14	0.2	0.27
Sunshine hours(City)	0.022	1	-0.37	0.35	-0.14	-0.37	0.22	-0.12	-0.44	0.26	-0.13
Life expectancy(years) (Country)	-0.32	-0.37	1	-0.57	0.66	0.72	-0.045	0.13	0.61	0.091	0.42
Pollution(Index score) (City)	0.46	0.35	-0.57	1	-0.51	-0.69	0.3	0.21	-0.66	-0.14	-0.28
Annual avg. hours worked	0.017	-0.14	0.66	-0.51	1	0.66	0.022	0.14	0.53	0.47	0.31
Happiness levels(Country)	-0.23	-0.37	0.72	-0.69	0.66	1	-0.14	0.033	0.81	0.45	0.3
Outdoor activities(City)	-0.077	0.22	-0.045	0.3	0.022	-0.14	1	0.53	-0.26	0.15	-0.12
Number of take out places(City)	0.36	-0.12	0.13	0.21	0.14	0.033	0.53	1	-0.09	0.017	0.093
water bottle cost	-0.14	-0.44	0.61	-0.66	0.53	0.81	-0.26	-0.09	1	0.29	0.36
obesity level	0.2	0.26	0.091	-0.14	0.47	0.45	0.15	0.017	0.29	1	-0.071
gym cost	0.27	-0.13	0.42	-0.28	0.31	0.3	-0.12	0.093	0.36	-0.071	1

In [65]:

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split,cross_val_score
from sklearn.metrics import mean_squared_error,r2_score
```

In [67]:

```
x = data["Happiness levels(Country)"].values.reshape(-1,1)
y = data["Life expectancy(years) (Country)"].values.reshape(-1,1)
```

In [68]:

```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.3)
```

In [69]:

```
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[69]:

```
LinearRegression()
```

In [70]:

```
y_pred = model.predict(X_test)
```

In [71]:

```
print("Training Accuracy :", model.score(X_train, y_train))
print("Testing Accuracy :", model.score(X_test, y_test))
```

```
Training Accuracy : 0.4571051671776636
Testing Accuracy : 0.6487255145665427
```

In [72]:

```
lin_reg = LinearRegression().fit(X_train,y_train)
```

In [73]:

```
lin_reg.coef_[0][0]
```

Out[73]:

```
3.6621780297879587
```

In [74]:

```
lin_reg.intercept_[0]
```

Out[74]:

```
54.86674932780668
```

In [75]:

```
print("{0}+{1}*Happiness Level".format(lin_reg.intercept_[0],lin_reg.coef_[0][0]))
```

```
54.86674932780668+3.6621780297879587*Happiness Level
```

In [76]:

```
y_pred = lin_reg.predict(X_test)
y_pred[0:10]
```

Out[76]:

```
array([[78.19482338],
       [80.28226485],
       [81.08794402],
       [80.9414569 ],
       [79.98929061],
       [76.36373436],
       [67.94072489],
       [80.758348  ],
       [80.28226485],
       [81.34429648]])
```

In [77]:

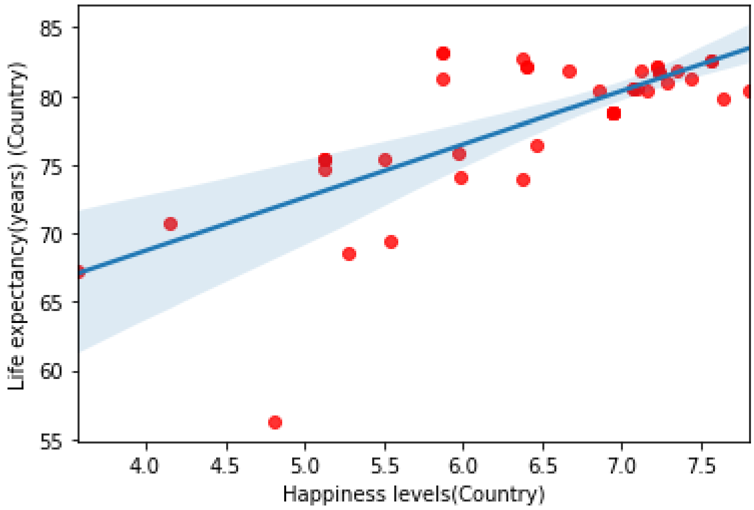
```
np.sqrt(mean_squared_error(y_test,y_pred))
```

Out[77]:

```
3.120839742094899
```

In [79]:

```
var = sns.regplot(data["Happiness levels(Country)"],data["Life expectancy(years) (Countr
```


Life expectancy(years) (Country)
Happiness levels(Country)