World Happiness Data 2024 Analysis

Features:

Country name: The country for which the data is reported.

Year: The year in which the data was collected.

Life Ladder: A measure of subjective well-being or life satisfaction on a scale where higher values generally indicate greater happiness.

Log GDP per capita: The logarithm of GDP per capita, reflecting economic prosperity and its impact on happiness.

Social support: A metric indicating the level of perceived social support or network available to individuals.

Healthy life expectancy at birth: The number of years a person is expected to live in good health from birth.

Freedom to make life choices: A measure of how free individuals feel in making life decisions.

Generosity: A metric reflecting the level of generosity or charitable giving in a country.

Perceptions of corruption: A measure of how corrupt the government is perceived to be, influencing trust and satisfaction.

Positive affect: The level of positive emotions such as joy and contentment experienced by individuals.

Negative affect: The level of negative emotions such as sadness and anxiety experienced by individuals.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings("ignore", category=FutureWarning)

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

In [2]:
```
df = pd.read_csv("World Happiness Report 2024.csv")
```

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

Out[3]:

	Country name	year	Life Ladder	Log GDP per capita	Social support	Healthy life expectancy at birth	Freedom to make life choices	Generosity	Perceptions of corruption	Positive affect	Negative affect
0	Afghanistan	2008	3.723590	7.350416	0.450662	50.500000	0.718114	0.164055	0.881686	0.414297	0.258195
1	Afghanistan	2009	4.401778	7.508646	0.552308	50.799999	0.678896	0.187297	0.850035	0.481421	0.237092
2	Afghanistan	2010	4.758381	7.613900	0.539075	51.099998	0.600127	0.117861	0.706766	0.516907	0.275324
3	Afghanistan	2011	3.831719	7.581259	0.521104	51.400002	0.495901	0.160098	0.731109	0.479835	0.267175
4	Afghanistan	2012	3.782938	7.660506	0.520637	51.700001	0.530935	0.234157	0.775620	0.613513	0.267919

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2363 entries, 0 to 2362
Data columns (total 11 columns):
 #   Column                            Non-Null Count  Dtype  
---  ------                            --------------  -----  
 0   Country name                      2363 non-null   object 
 1   year                              2363 non-null   int64  
 2   Life Ladder                       2363 non-null   float64
 3   Log GDP per capita                2335 non-null   float64
 4   Social support                    2350 non-null   float64
 5   Healthy life expectancy at birth  2300 non-null   float64
 6   Freedom to make life choices      2327 non-null   float64
 7   Generosity                        2282 non-null   float64
 8   Perceptions of corruption         2238 non-null   float64
 9   Positive affect                   2339 non-null   float64
 10  Negative affect                   2347 non-null   float64
dtypes: float64(9), int64(1), object(1)
memory usage: 203.2+ KB
```

In [5]:
```
df.isna().sum()
```

Out[5]:
```
Country name                          0
year                                  0
Life Ladder                           0
Log GDP per capita                   28
Social support                       13
Healthy life expectancy at birth     63
Freedom to make life choices         36
Generosity                           81
Perceptions of corruption           125
Positive affect                      24
Negative affect                      16
dtype: int64
```

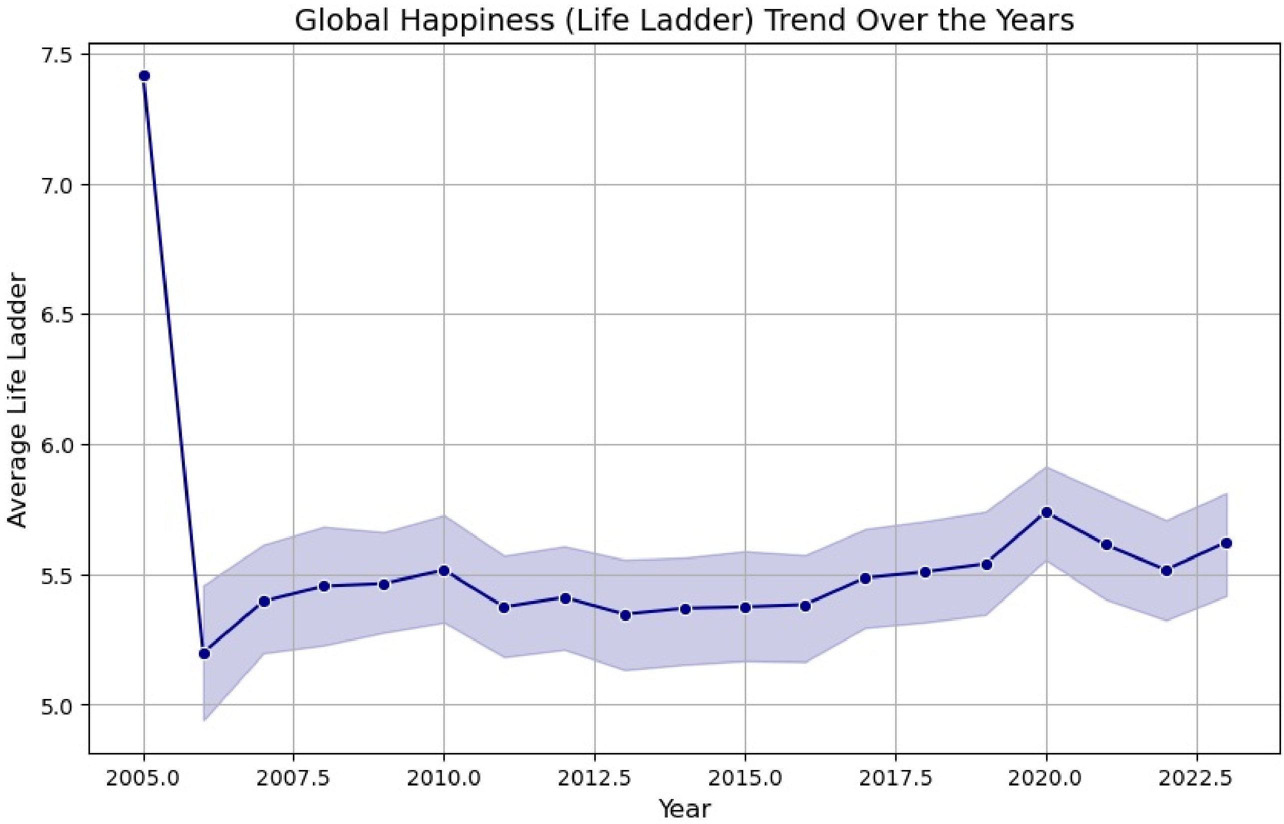
In [6]:
```
df = df.dropna()
```

In [7]:
```
df.isna().sum()
```

Out[7]:
```
Country name                        0
year                                0
Life Ladder                         0
Log GDP per capita                  0
Social support                      0
Healthy life expectancy at birth    0
Freedom to make life choices        0
Generosity                          0
Perceptions of corruption           0
Positive affect                     0
Negative affect                     0
dtype: int64
```

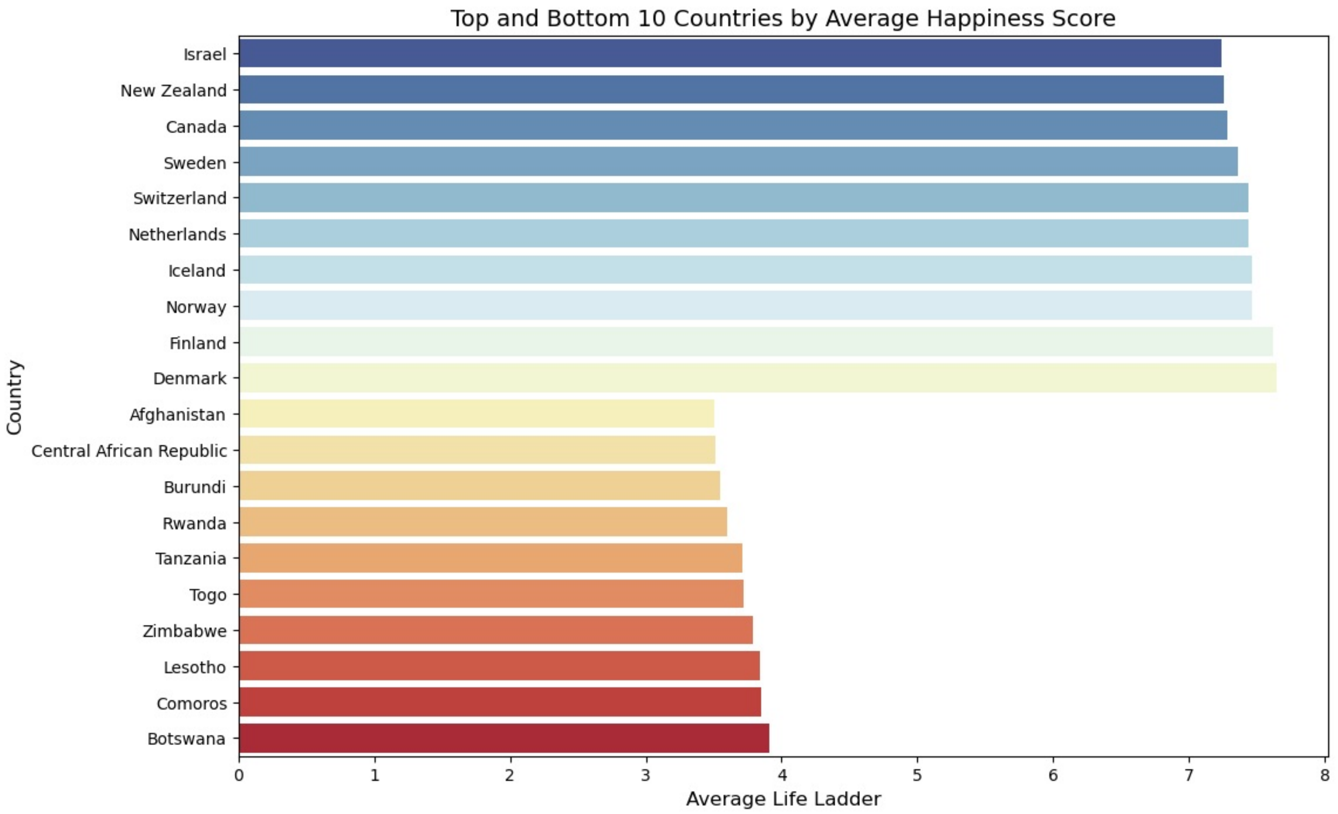
How has the global happiness score (Life Ladder) trended over the years?

In [8]:
```
plt.figure(figsize=(10,6))
sns.lineplot(data=df, x='year', y='Life Ladder', marker='o', color='darkblue')

plt.title('Global Happiness (Life Ladder) Trend Over the Years', fontsize=14)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Average Life Ladder', fontsize=12)
plt.grid(True)
plt.show()
```



Which countries have the highest and lowest average happiness scores?

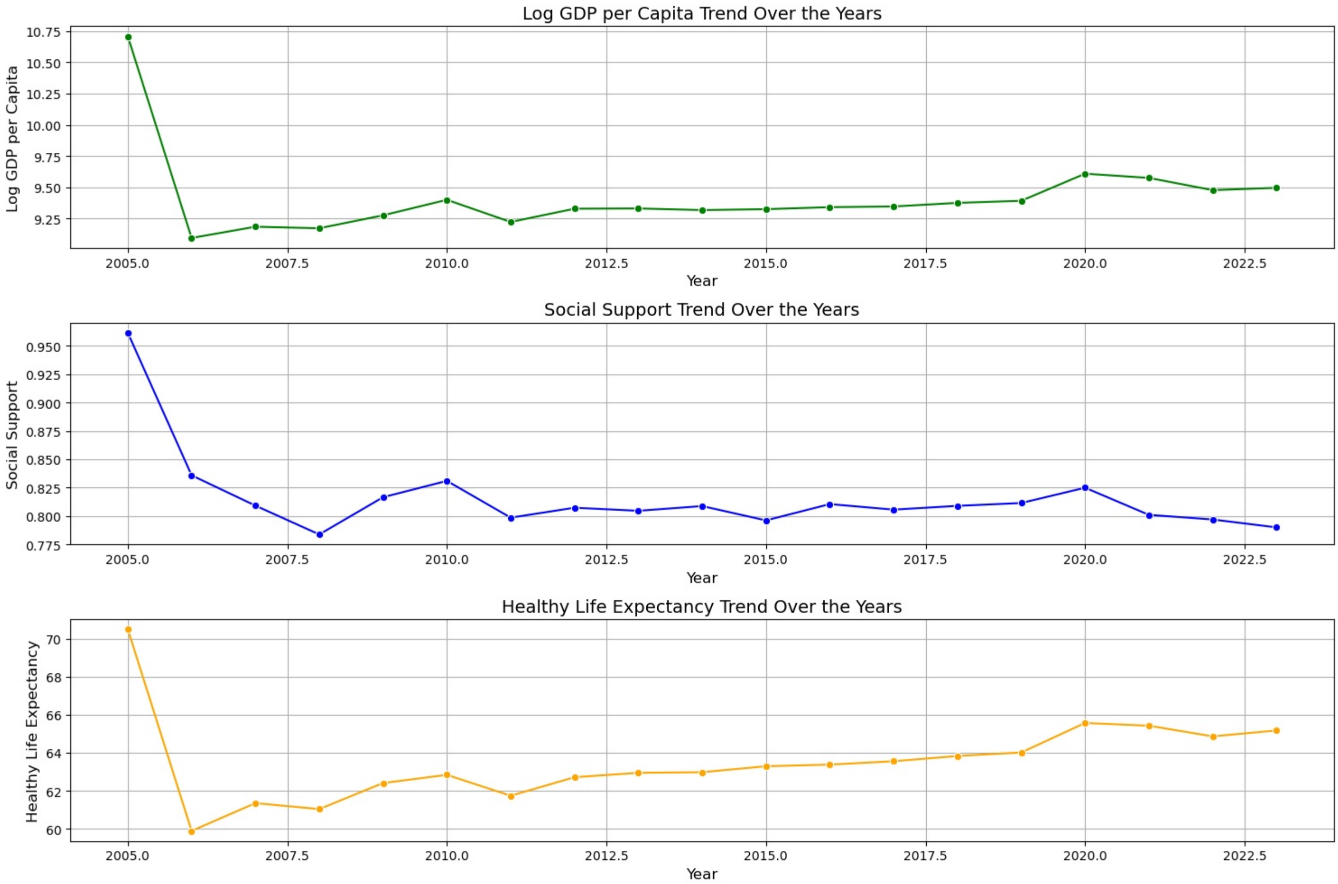
In [9]:
```
# Average Life Ladder by country
country_ladder_avg = df.groupby('Country name')['Life Ladder'].mean().sort_values()

# Top and Bottom 10 countries
top_10_countries = country_ladder_avg.tail(10)
bottom_10_countries = country_ladder_avg.head(10)

top_bottom_countries = pd.concat([top_10_countries, bottom_10_countries])

plt.figure(figsize=(12, 8))
sns.barplot(x=top_bottom_countries.values, y=top_bottom_countries.index, palette='RdYlBu_r')

plt.title('Top and Bottom 10 Countries by Average Happiness Score', fontsize=14)
plt.xlabel('Average Life Ladder', fontsize=12)
plt.ylabel('Country', fontsize=12)
plt.show()
```



How have Log GDP per capita, Social support, and Healthy life expectancy at birth trended over the years?

In [10]:

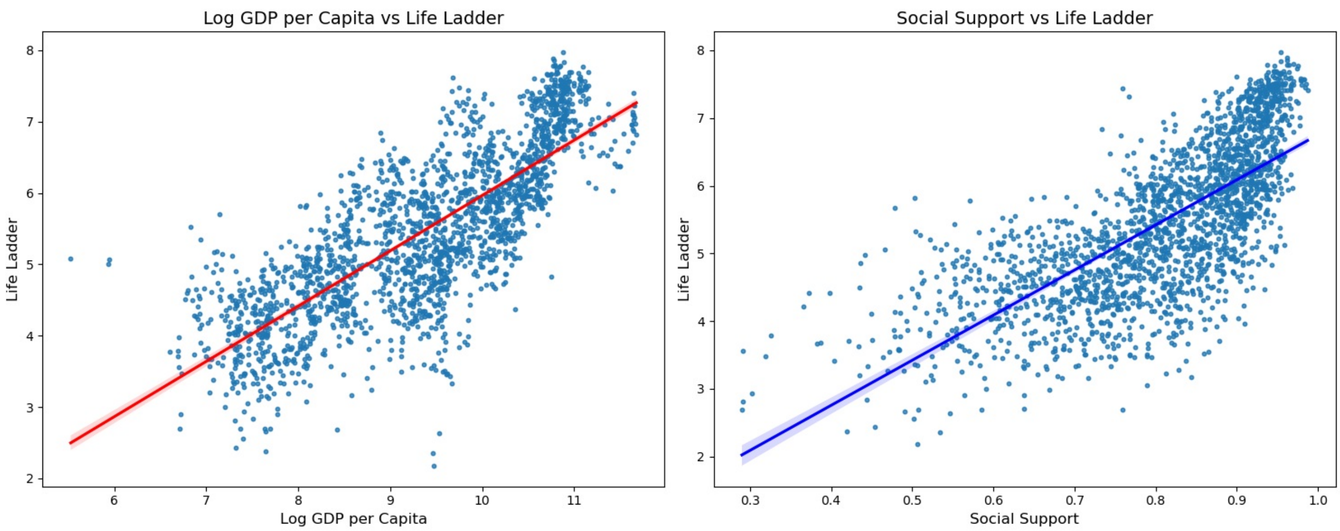
```
plt.figure(figsize=(15, 10))

# Plot for Log GDP per capita
plt.subplot(3, 1, 1)
sns.lineplot(data=df, x='year', y='Log GDP per capita', errorbar=None, marker='o', color='green')
plt.title('Log GDP per Capita Trend Over the Years', fontsize=14)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Log GDP per Capita', fontsize=12)
plt.grid(True)

# Plot for Social support
plt.subplot(3, 1, 2)
sns.lineplot(data=df, x='year', y='Social support', errorbar=None, marker='o', color='blue')
plt.title('Social Support Trend Over the Years', fontsize=14)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Social Support', fontsize=12)
plt.grid(True)

# Plot for Healthy life expectancy at birth
plt.subplot(3, 1, 3)
sns.lineplot(data=df, x='year', y='Healthy life expectancy at birth', errorbar=None, marker='o', color='orange'
plt.title('Healthy Life Expectancy Trend Over the Years', fontsize=14)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Healthy Life Expectancy', fontsize=12)
plt.grid(True)

plt.tight_layout()
plt.show()
```



How do Log GDP per capita and Social support correlate with Life Ladder, and what do their relationships look like?

In [11]:

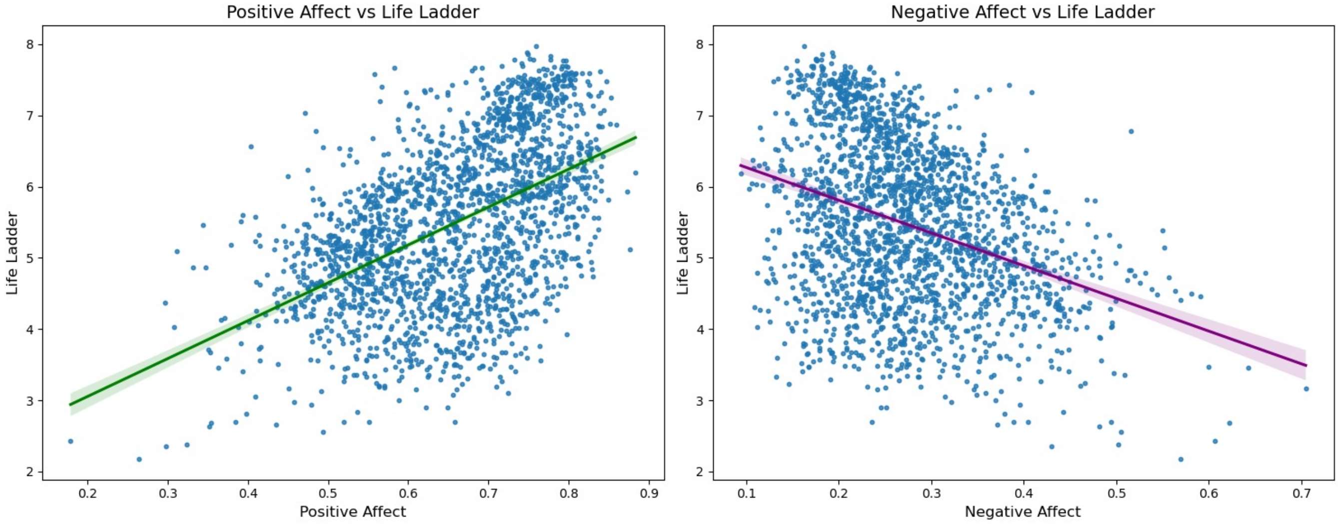
```python
plt.figure(figsize=(15, 6))

# Plot for Log GDP per capita vs Life Ladder
plt.subplot(1, 2, 1)
sns.regplot(data=df, x='Log GDP per capita', y='Life Ladder', scatter_kws={'s':10}, line_kws={'color':'red'})
plt.title('Log GDP per Capita vs Life Ladder', fontsize=14)
plt.xlabel('Log GDP per Capita', fontsize=12)
plt.ylabel('Life Ladder', fontsize=12)

# Plot for Social support vs Life Ladder
plt.subplot(1, 2, 2)
sns.regplot(data=df, x='Social support', y='Life Ladder', scatter_kws={'s':10}, line_kws={'color':'blue'})
plt.title('Social Support vs Life Ladder', fontsize=14)
plt.xlabel('Social Support', fontsize=12)
plt.ylabel('Life Ladder', fontsize=12)

plt.tight_layout()
plt.show()
```



How do Positive affect and Negative affect relate to the Life

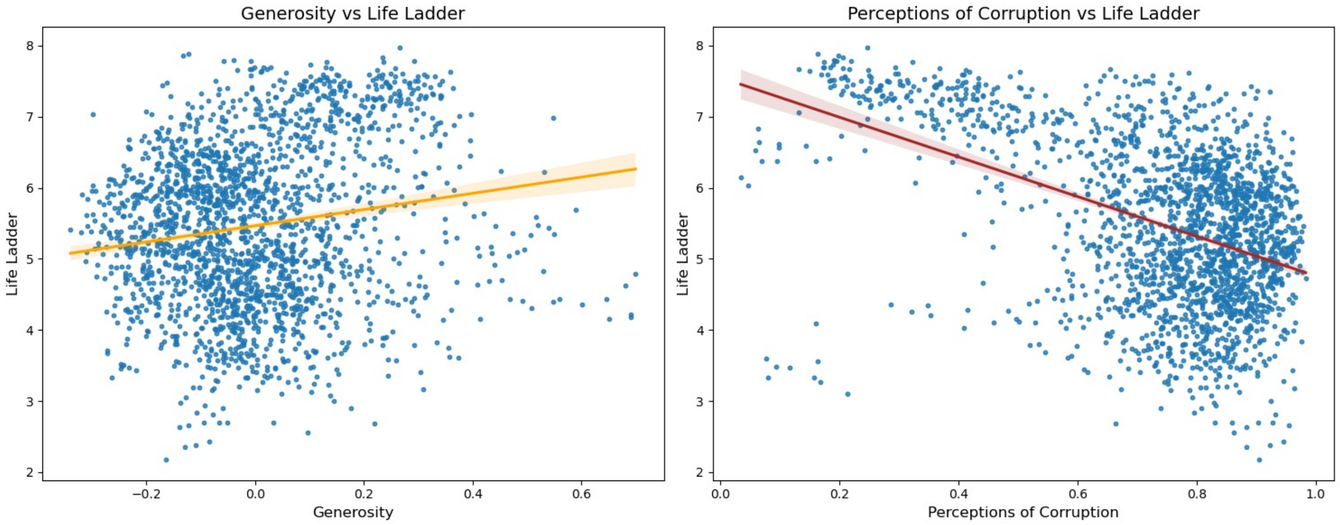
Ladder?

In [12]:
```
plt.figure(figsize=(15, 6))

# Plot for Positive affect vs Life Ladder
plt.subplot(1, 2, 1)
sns.regplot(data=df, x='Positive affect', y='Life Ladder', scatter_kws={'s':10}, line_kws={'color':'green'})
plt.title('Positive Affect vs Life Ladder', fontsize=14)
plt.xlabel('Positive Affect', fontsize=12)
plt.ylabel('Life Ladder', fontsize=12)

# Plot for Negative affect vs Life Ladder
plt.subplot(1, 2, 2)
sns.regplot(data=df, x='Negative affect', y='Life Ladder', scatter_kws={'s':10}, line_kws={'color':'purple'})
plt.title('Negative Affect vs Life Ladder', fontsize=14)
plt.xlabel('Negative Affect', fontsize=12)
plt.ylabel('Life Ladder', fontsize=12)

plt.tight_layout()
plt.show()
```



How do Generosity and Perceptions of corruption relate to the Life Ladder?

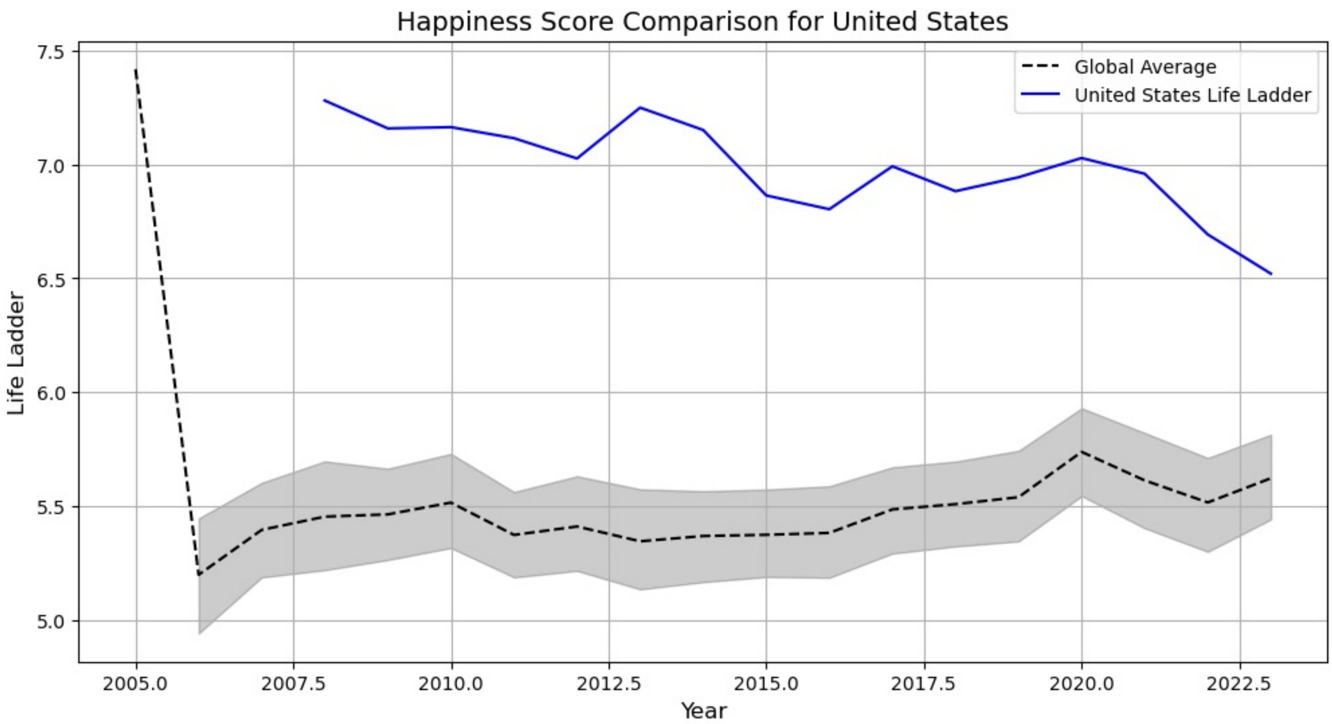
In [13]:
```
plt.figure(figsize=(15, 6))

# Plot for Generosity vs Life Ladder
plt.subplot(1, 2, 1)
sns.regplot(data=df, x='Generosity', y='Life Ladder', scatter_kws={'s':10}, line_kws={'color':'orange'})
plt.title('Generosity vs Life Ladder', fontsize=14)
plt.xlabel('Generosity', fontsize=12)
plt.ylabel('Life Ladder', fontsize=12)

# Plot for Perceptions of corruption vs Life Ladder
plt.subplot(1, 2, 2)
sns.regplot(data=df, x='Perceptions of corruption', y='Life Ladder', scatter_kws={'s':10}, line_kws={'color':'b
plt.title('Perceptions of Corruption vs Life Ladder', fontsize=14)
plt.xlabel('Perceptions of Corruption', fontsize=12)
plt.ylabel('Life Ladder', fontsize=12)

plt.tight_layout()
plt.show()
```



1. Country Comparison

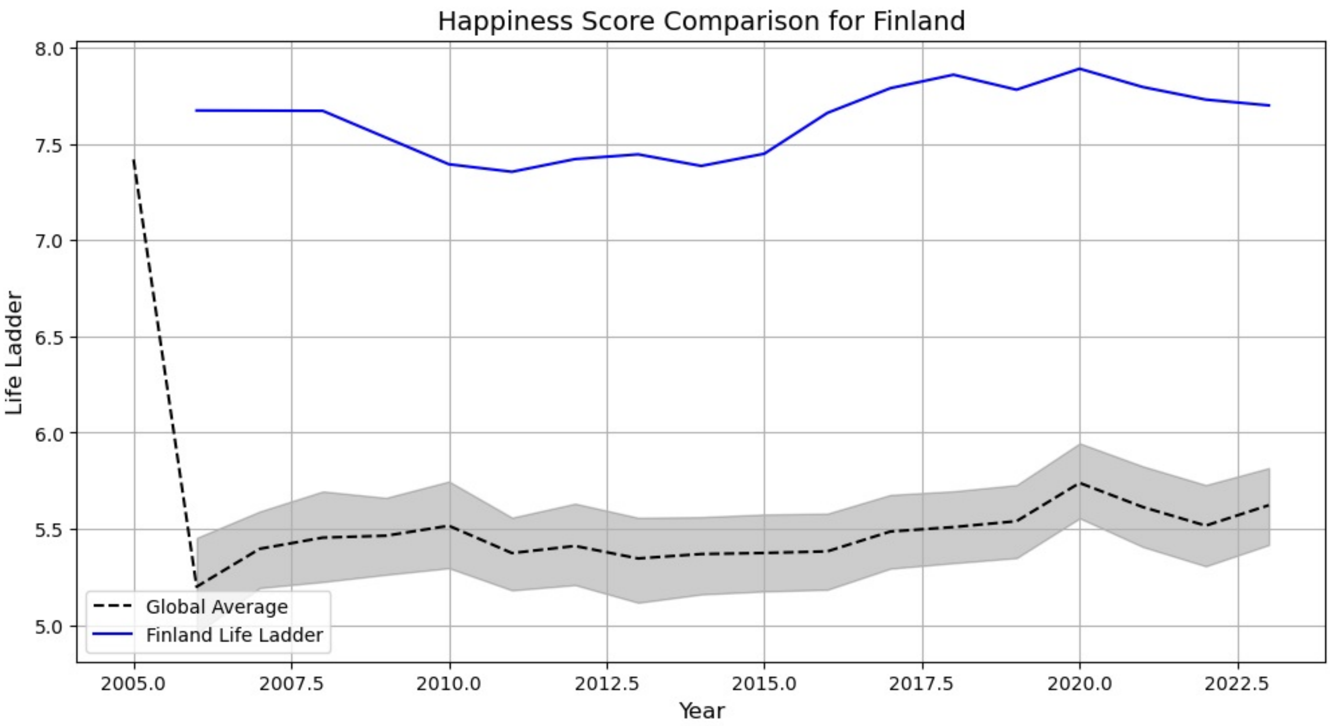
To compare the happiness score of a specific country to the global average:

In [14]:
```python
# For US
country = 'United States'
df_country = df[df['Country name'] == country]

# Global average Life Ladder
global_avg_life_ladder = df['Life Ladder'].mean()

plt.figure(figsize=(12, 6))
sns.lineplot(data=df, x='year', y='Life Ladder', label='Global Average', color='black', linestyle='--')
sns.lineplot(data=df_country, x='year', y='Life Ladder', label=f'{country} Life Ladder', color='blue')
plt.title(f'Happiness Score Comparison for {country}', fontsize=14)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Life Ladder', fontsize=12)
plt.legend()
plt.grid(True)
plt.show()
```

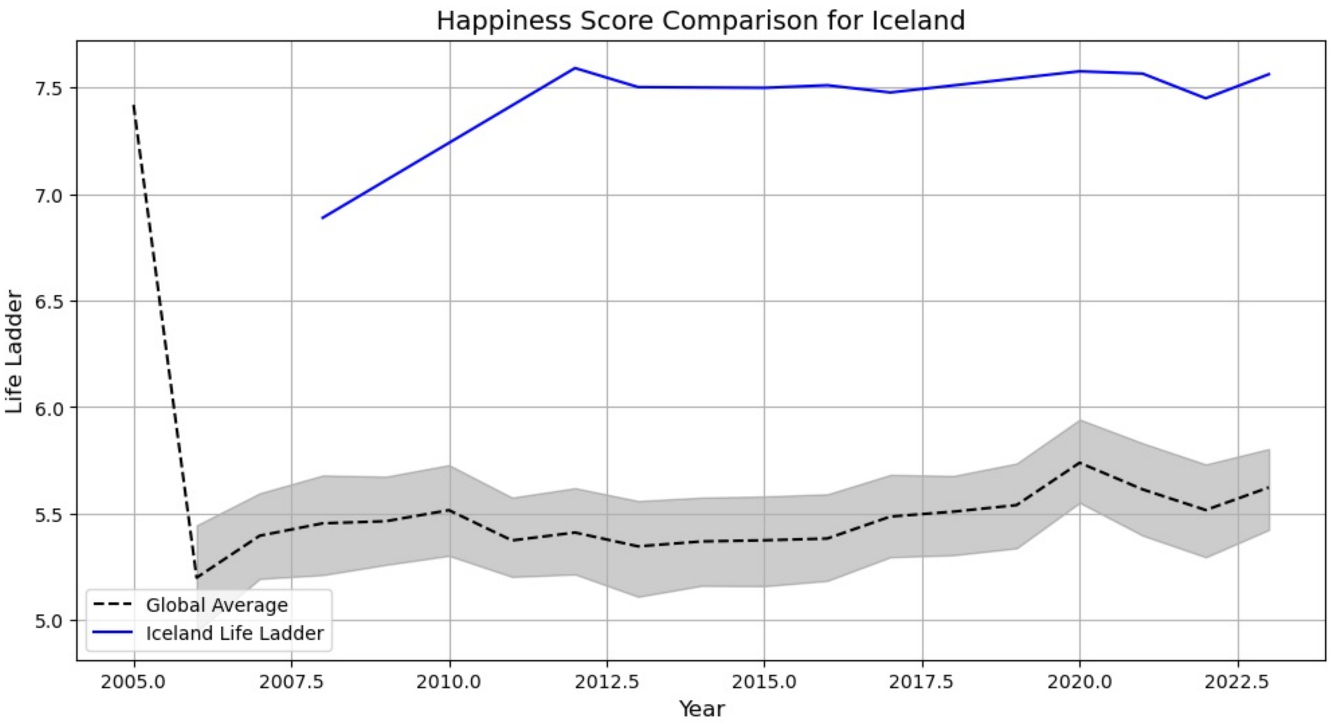


In [15]:
```python
# For Finland
country = 'Finland'
df_country = df[df['Country name'] == country]

# Global average Life Ladder
global_avg_life_ladder = df['Life Ladder'].mean()

plt.figure(figsize=(12, 6))
sns.lineplot(data=df, x='year', y='Life Ladder', label='Global Average', color='black', linestyle='--')
sns.lineplot(data=df_country, x='year', y='Life Ladder', label=f'{country} Life Ladder', color='blue')
```

```python
plt.title(f'Happiness Score Comparison for {country}', fontsize=14)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Life Ladder', fontsize=12)
plt.legend()
plt.grid(True)
plt.show()
```

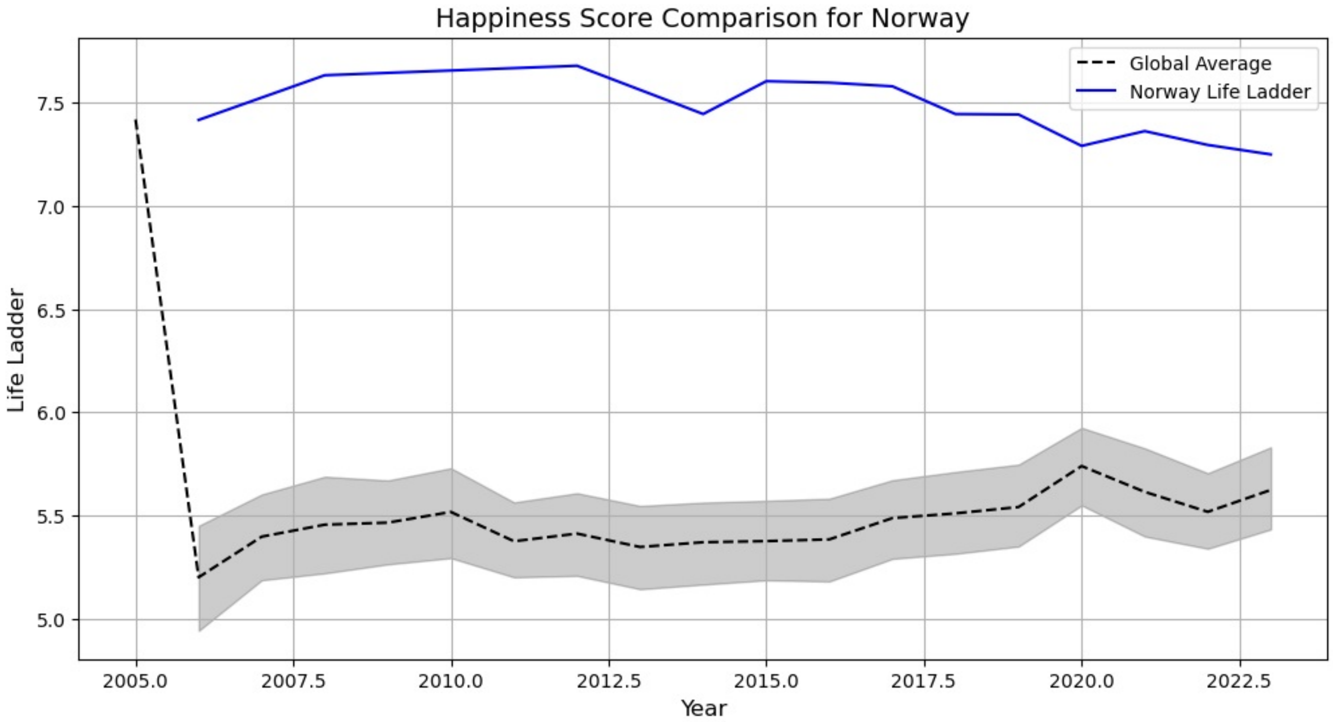


In [16]:
```python
# For Iceland
country = 'Iceland'
df_country = df[df['Country name'] == country]

# Global average Life Ladder
global_avg_life_ladder = df['Life Ladder'].mean()

plt.figure(figsize=(12, 6))
sns.lineplot(data=df, x='year', y='Life Ladder', label='Global Average', color='black', linestyle='--')
sns.lineplot(data=df_country, x='year', y='Life Ladder', label=f'{country} Life Ladder', color='blue')
plt.title(f'Happiness Score Comparison for {country}', fontsize=14)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Life Ladder', fontsize=12)
plt.legend()
plt.grid(True)
plt.show()
```



In [17]:
```python
# For Norway
country = 'Norway'
df_country = df[df['Country name'] == country]
```

```python
# Global average Life Ladder
global_avg_life_ladder = df['Life Ladder'].mean()

plt.figure(figsize=(12, 6))
sns.lineplot(data=df, x='year', y='Life Ladder', label='Global Average', color='black', linestyle='--')
sns.lineplot(data=df_country, x='year', y='Life Ladder', label=f'{country} Life Ladder', color='blue')
plt.title(f'Happiness Score Comparison for {country}', fontsize=14)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Life Ladder', fontsize=12)
plt.legend()
plt.grid(True)
plt.show()
```



2. Feature Impact Analysis

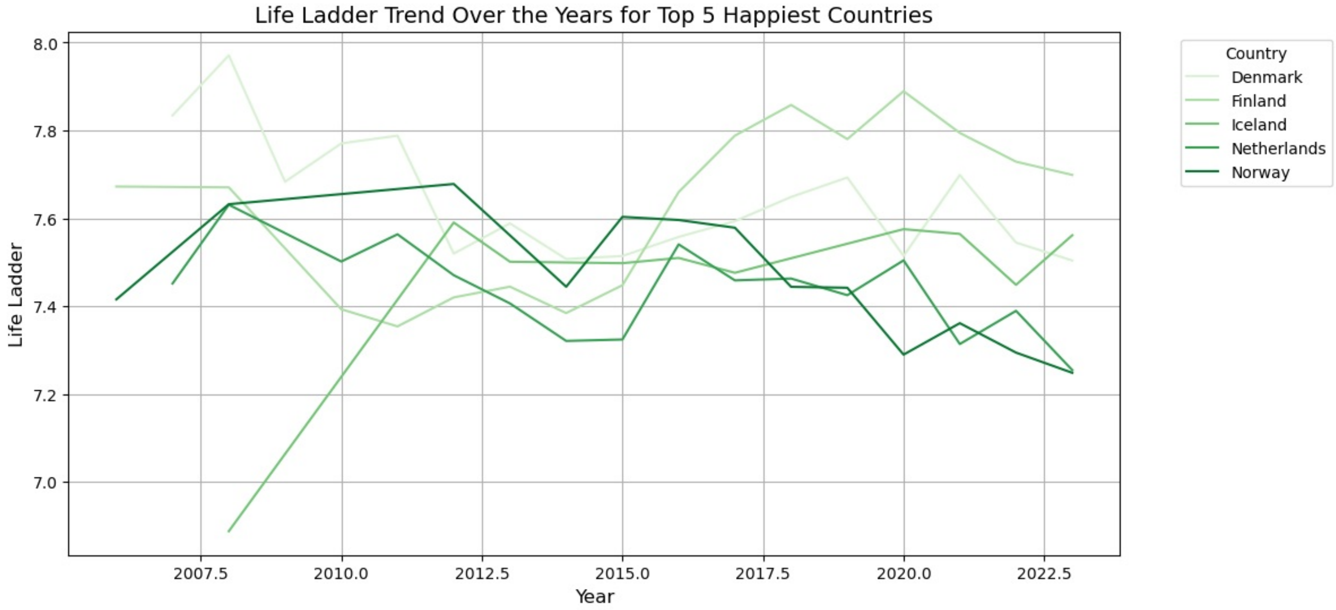
To analyze the impact of Healthy life expectancy at birth:

In [18]:
```python
plt.figure(figsize=(12, 6))
sns.scatterplot(data=df, x='Healthy life expectancy at birth', y='Life Ladder', hue='year', palette='PiYG')
plt.title('Healthy Life Expectancy vs Life Ladder', fontsize=14)
plt.xlabel('Healthy Life Expectancy at Birth', fontsize=12)
plt.ylabel('Life Ladder', fontsize=12)
plt.legend(title='Year', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```

3. Time-Based Patterns

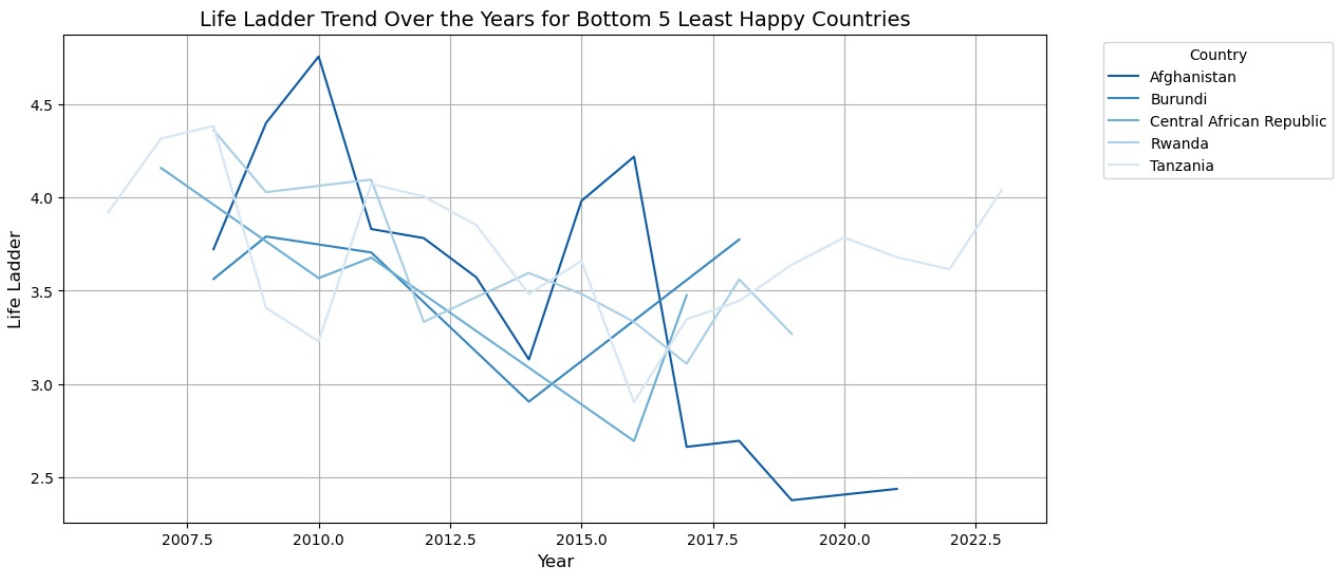
Top 5 Happy Countries

In [19]:
```
# Average Life Ladder score for each country
top_countries = df.groupby('Country name')['Life Ladder'].mean().nlargest(5).index

df_top_countries = df[df['Country name'].isin(top_countries)]

plt.figure(figsize=(12, 6))
sns.lineplot(data=df_top_countries, x='year', y='Life Ladder', hue='Country name', palette='Greens', errorbar=N
plt.title('Life Ladder Trend Over the Years for Top 5 Happiest Countries', fontsize=14)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Life Ladder', fontsize=12)
plt.legend(title='Country', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.grid(True)
plt.show()
```



Bottom 5 Happy Countries

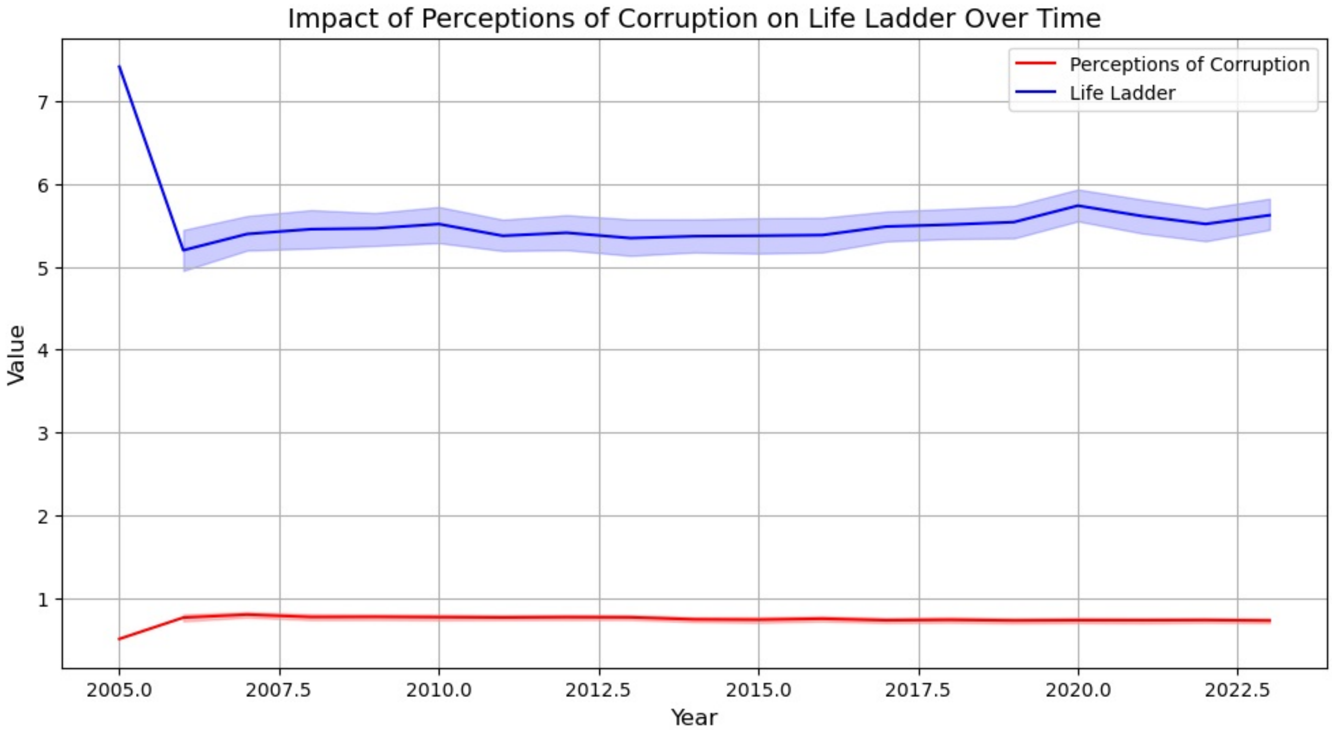
In [20]:
```
# Average Life Ladder score for each country
bottom_countries = df.groupby('Country name')['Life Ladder'].mean().nsmallest(5).index

df_bottom_countries = df[df['Country name'].isin(bottom_countries)]

plt.figure(figsize=(12, 6))
sns.lineplot(data=df_bottom_countries, x='year', y='Life Ladder', hue='Country name', palette='Blues_r', errorb
plt.title('Life Ladder Trend Over the Years for Bottom 5 Least Happy Countries', fontsize=14)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Life Ladder', fontsize=12)
plt.legend(title='Country', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.grid(True)
plt.show()
```



4. Longitudinal Analysis

To analyze how Perceptions of corruption influences Life Ladder over time:

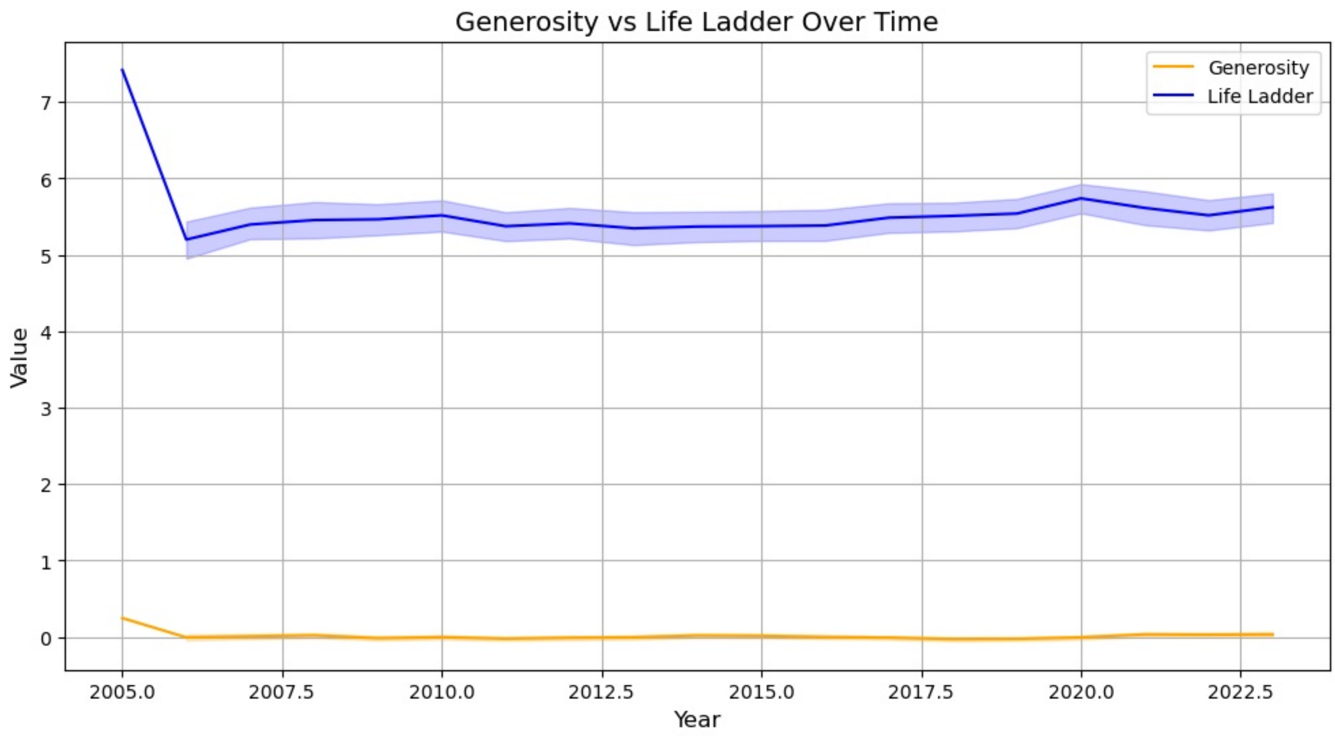
In [21]:
```
plt.figure(figsize=(12, 6))
sns.lineplot(data=df, x='year', y='Perceptions of corruption', label='Perceptions of Corruption', color='red')
sns.lineplot(data=df, x='year', y='Life Ladder', label='Life Ladder', color='blue')
plt.title('Impact of Perceptions of Corruption on Life Ladder Over Time', fontsize=14)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Value', fontsize=12)
plt.legend()
plt.grid(True)
plt.show()
```



5. Generosity and Happiness

To explore how Generosity correlates with happiness over time:

In [22]:
```
plt.figure(figsize=(12, 6))
sns.lineplot(data=df, x='year', y='Generosity', label='Generosity', color='orange')
sns.lineplot(data=df, x='year', y='Life Ladder', label='Life Ladder', color='blue')
plt.title('Generosity vs Life Ladder Over Time', fontsize=14)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Value', fontsize=12)
plt.legend()
plt.grid(True)
plt.show()
```



In [23]:
```
df.head()
```

Out[23]:

	Country name	year	Life Ladder	Log GDP per capita	Social support	Healthy life expectancy at birth	Freedom to make life choices	Generosity	Perceptions of corruption	Positive affect	Negative affect
0	Afghanistan	2008	3.723590	7.350416	0.450662	50.500000	0.718114	0.164055	0.881686	0.414297	0.258195
1	Afghanistan	2009	4.401778	7.508646	0.552308	50.799999	0.678896	0.187297	0.850035	0.481421	0.237092
2	Afghanistan	2010	4.758381	7.613900	0.539075	51.099998	0.600127	0.117861	0.706766	0.516907	0.275324
3	Afghanistan	2011	3.831719	7.581259	0.521104	51.400002	0.495901	0.160098	0.731109	0.479835	0.267175
4	Afghanistan	2012	3.782938	7.660506	0.520637	51.700001	0.530935	0.234157	0.775620	0.613513	0.267919

LETS TRY TO PREDICT LIFE LADDER

In [24]:
```
df_features = df.drop(columns=['Country name', 'year'])
```

In [25]:
```
# Defining features and target variable
X = df_features.drop(columns=['Life Ladder'])
y = df_features['Life Ladder']
```

In [26]:
```
# Spliting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [27]:
```
# Standardize the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

In [28]:
```
# Initializing and training the model
model = LinearRegression()
model.fit(X_train_scaled, y_train)
```

Out[28]:

```
▾ LinearRegression
LinearRegression()
```

In [29]:
```
# Prediction
y_pred = model.predict(X_test_scaled)
```

In [30]:
```
# Evaluating the model
mse = mean_squared_error(y_test, y_pred)
rmse = mse ** 0.5
r2 = r2_score(y_test, y_pred)

print(f"RMSE: {rmse:.4f}")
print(f"R² Score: {r2:.4f}")
```

```
RMSE: 0.5586
R² Score: 0.7350
```

In [31]:

```python
# Predicting values on the test set
y_pred = model.predict(X_test_scaled)

# DataFrame with actual and predicted values
results_df = pd.DataFrame({
    'Actual Life Ladder': y_test.values,
    'Predicted Life Ladder': y_pred,
    'Difference': y_test.values - y_pred
})

results_df.head()
```

Out[31]:

	Actual Life Ladder	Predicted Life Ladder	Difference
0	5.317194	5.091727	0.225468
1	5.694870	5.961050	-0.266180
2	5.145833	4.289176	0.856657
3	7.432132	6.413649	1.018483
4	6.012740	5.678136	0.334604

Thank You!