Employee Wellness, Performance, and Retention Analysis

In [1]:
```
import os;
os.listdir ('/kaggle/input/')
```

Out[1]:
```
['wellness2', 'wellness']
```

IMPORTING THE DATA SET

In [2]:
```
#import library
import pandas as pd

#Read each sheet into a Dataframe

df_sheet1 = pd.read_excel('/kaggle/input/wellness/Vacation and wellness.xlsx', sheet_name = 'Vacation n Wellness 2022')
df_sheet2 = pd.read_excel('/kaggle/input/wellness/Vacation and wellness.xlsx', sheet_name = 'Vacation n Wellness 2023')
```

Merged sheets and Display the first few rows

In [3]:
```
#Merged the sheets using a full outer join

merged_df = pd.merge (df_sheet1, df_sheet2, on = 'Employee ID', how = 'outer')

#Display the first few rows of the merged Dataframe

merged_df.head()
```

Out[3]:

	Employee ID	Gender_x	Hire Date_x	Position Title_x	Grade_x	Division Classification_x	Company_x	Career Stream_x	Event Date_x	Employee Status_x	...	Event	Term
0	24	Woman	2003-05-26	Mgr Client Services & Administration	P7	MI-ISS	Mackenzie Financial Corporation	Professional	2020-02-19	Terminated	...	Termination	202
1	101	Woman	1982-08-09	Snr Mgr Client Act Mgt & Special Sup	L8	IGM-CSO	Investors Group Financial Services	Leader	2022-10-16	Active	...	Data Change	
2	104	Woman	1982-12-13	Loss Mitigation Analyst	P5	IG-InsMtgBnk	Investors Group Investment Management	Professional	2022-03-01	Active	...	Data Change	202
3	161	Man	1980-10-27	Director Real Estate Asset Mgmt	P9	IG-IGIM	Investors Group Investment Management	Professional	2021-01-01	Terminated	...	Termination	202
4	311	Woman	1977-07-04	Snr Mgr Learning & Communication	L8	IGM-CSO	Investors Group Financial Services	Leader	2020-02-12	Terminated	...	Termination	202

5 rows × 46 columns

DISPLAY BASIC INFORMATION AND STATISTICS OF THE DATAFRAME

In [4]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns


# Display basic info and statistics of the DataFrame

print(merged_df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6929 entries, 0 to 6928
Data columns (total 46 columns):
 #   Column                         Non-Null Count  Dtype         
---  ------                         --------------  -----         
 0   Employee ID                    6929 non-null   int64         
 1   Gender_x                       6044 non-null   object        
 2   Hire Date_x                    6044 non-null   datetime64[ns]
 3   Position Title_x               6043 non-null   object        
 4   Grade_x                        6044 non-null   object        
 5   Division Classification_x      6044 non-null   object        
 6   Company_x                      6044 non-null   object        
 7   Career Stream_x                6044 non-null   object        
 8   Event Date_x                   6044 non-null   datetime64[ns]
 9   Employee Status_x              6044 non-null   object        
 10  Termination Date_x             1968 non-null   datetime64[ns]
 11  STIP Total Target_x            5747 non-null   object        
 12  STIP Total Max_x               5747 non-null   object        
 13  Vacation Annual Entitlement_x  6043 non-null   float64       
 14  Age_x                          6044 non-null   float64       
 15  Wellness - Used_x              6044 non-null   float64       
 16  Wellness - Accrued_x           6044 non-null   float64       
 17  Wellness - %                   6044 non-null   float64       
 18  Wellness Used round up         6044 non-null   object        
 19  Vacation - Used_x              6044 non-null   float64       
 20  Vacation - Accrued_x           6044 non-null   float64       
 21  Vacation - %                   6044 non-null   float64       
 22  Vacation Used round up         6044 non-null   object        
 23  Age_y                          6916 non-null   float64       
 24  Gender_y                       6916 non-null   object        
 25  Employee Status_y              6916 non-null   object        
 26  Company_y                      6916 non-null   object        
 27  Division Classification_y      6916 non-null   object        
 28  Hire Date_y                    6916 non-null   datetime64[ns]
 29  Position Title_y               6916 non-null   object        
 30  Grade_y                        6916 non-null   object        
 31  Career Stream_y                6916 non-null   object        
 32  STIP Total Target_y            6655 non-null   object        
 33  STIP Total Max_y               6655 non-null   object        
 34  Vacation Annual Entitlement_y  6915 non-null   float64       
 35  Event Date_y                   6916 non-null   datetime64[ns]
 36  Event                          6916 non-null   object        
 37  Termination Date_y             2759 non-null   datetime64[ns]
 38  Wellness - Used_y              6916 non-null   float64       
 39  Wellness - Accrued_y           6916 non-null   float64       
 40  Wellness Used- %               6916 non-null   float64       
 41  Wellness-round up              6916 non-null   object        
 42  Vacation - Used_y              6916 non-null   float64       
 43  Vacation - Accrued_y           6916 non-null   float64       
 44  Vacation Used - %              6916 non-null   float64       
 45  Vacation-round up              6916 non-null   object        
dtypes: datetime64[ns](6), float64(16), int64(1), object(23)
memory usage: 2.4+ MB
None
```

In [5]:

```
# Check for missing values
print(merged_df.isnull().sum())
```

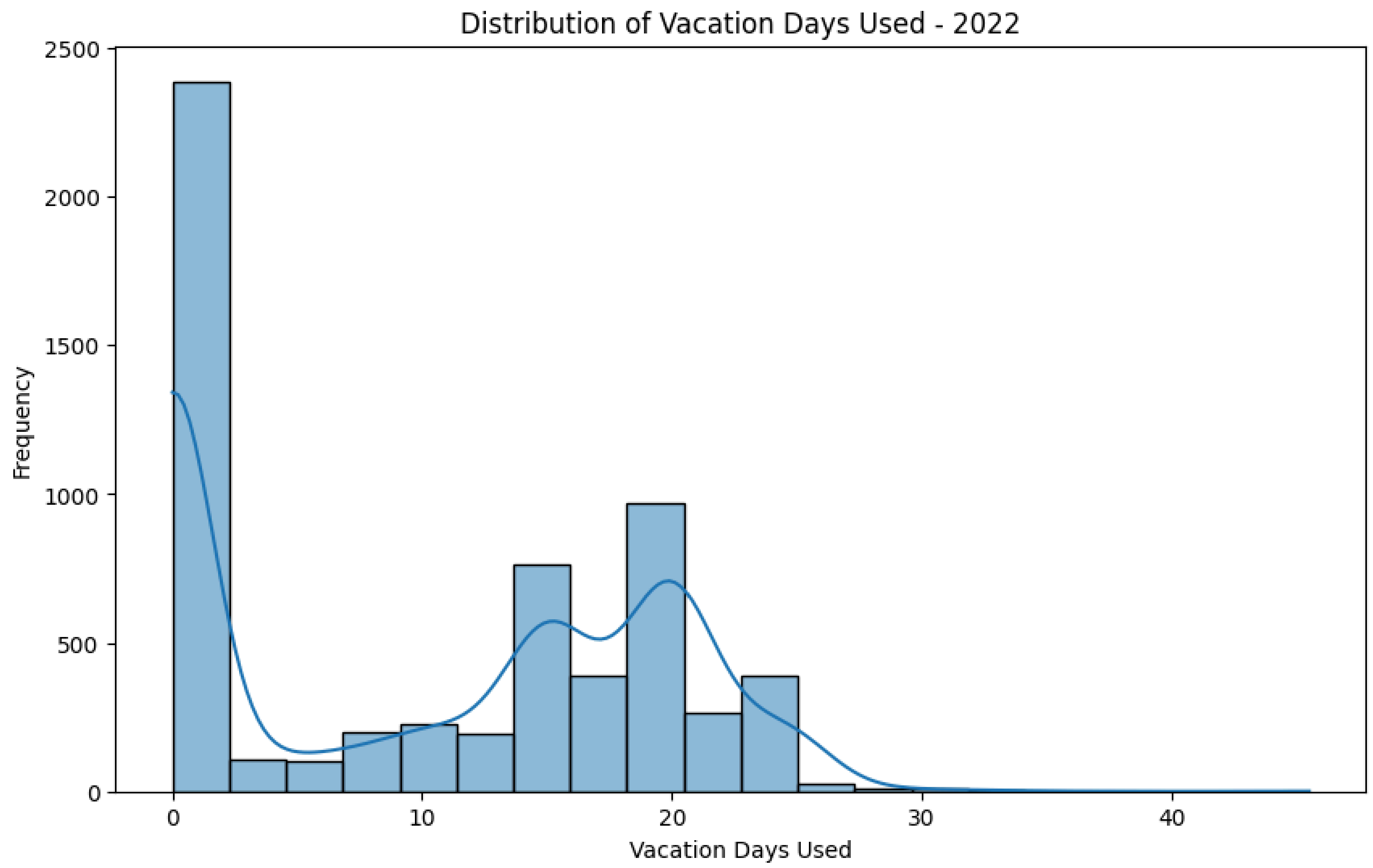
```
Employee ID                          0
Gender_x                           885
Hire Date_x                        885
Position Title_x                   886
Grade_x                            885
Division Classification_x          885
Company_x                          885
Career Stream_x                    885
Event Date_x                       885
Employee Status_x                  885
Termination Date_x                4961
STIP Total Target_x               1182
STIP Total Max_x                  1182
Vacation Annual Entitlement_x      886
Age_x                              885
Wellness - Used_x                  885
Wellness - Accrued_x               885
Wellness - %                       885
Wellness Used round up             885
Vacation - Used_x                  885
Vacation - Accrued_x               885
Vacation - %                       885
Vacation Used round up             885
Age_y                               13
Gender_y                            13
Employee Status_y                   13
Company_y                           13
Division Classification_y           13
Hire Date_y                         13
Position Title_y                    13
Grade_y                             13
Career Stream_y                     13
STIP Total Target_y                274
STIP Total Max_y                   274
Vacation Annual Entitlement_y       14
Event Date_y                        13
Event                               13
Termination Date_y                4170
Wellness - Used_y                   13
Wellness - Accrued_y                13
Wellness Used- %                    13
Wellness-round up                   13
Vacation - Used_y                   13
Vacation - Accrued_y                13
Vacation Used - %                   13
Vacation-round up                   13
dtype: int64
```

DISTRIBUTION OF VACATION DAYS USED

In [6]:

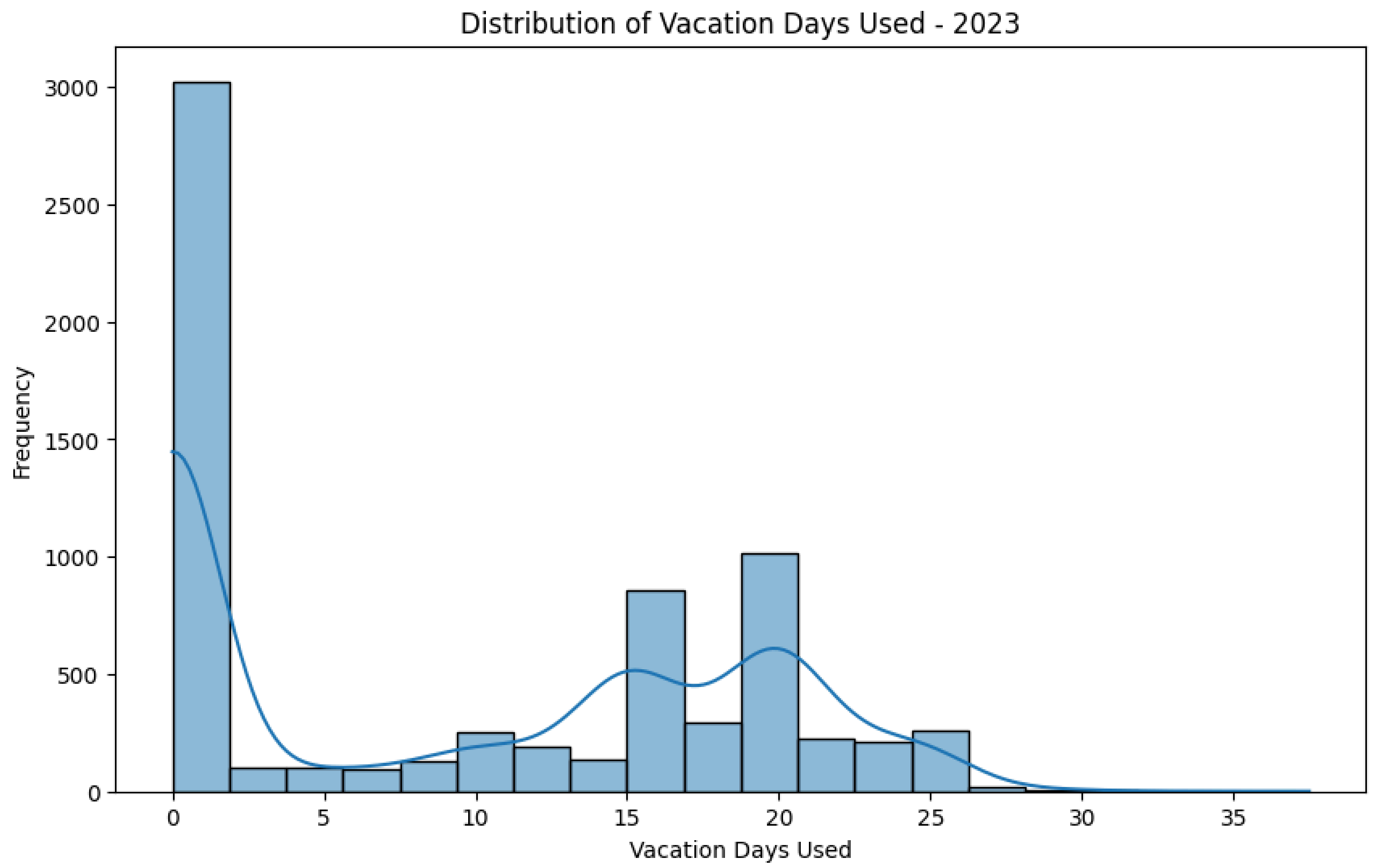
```
import matplotlib.pyplot as plt
import seaborn as sns

# Plot distribution of vacation days used for 2022
plt.figure(figsize=(10, 6))
sns.histplot(merged_df['Vacation - Used_x'], bins=20, kde=True)
plt.title('Distribution of Vacation Days Used - 2022')
plt.xlabel('Vacation Days Used')
plt.ylabel('Frequency')
plt.show()
```

```
/opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and
will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



In [7]:
```
# Plot distribution of vacation days used for 2023
plt.figure(figsize=(10, 6))
sns.histplot(merged_df['Vacation - Used_y'], bins=20, kde=True)
plt.title('Distribution of Vacation Days Used - 2023')
plt.xlabel('Vacation Days Used')
plt.ylabel('Frequency')
plt.show()
```

```
/opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and
will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



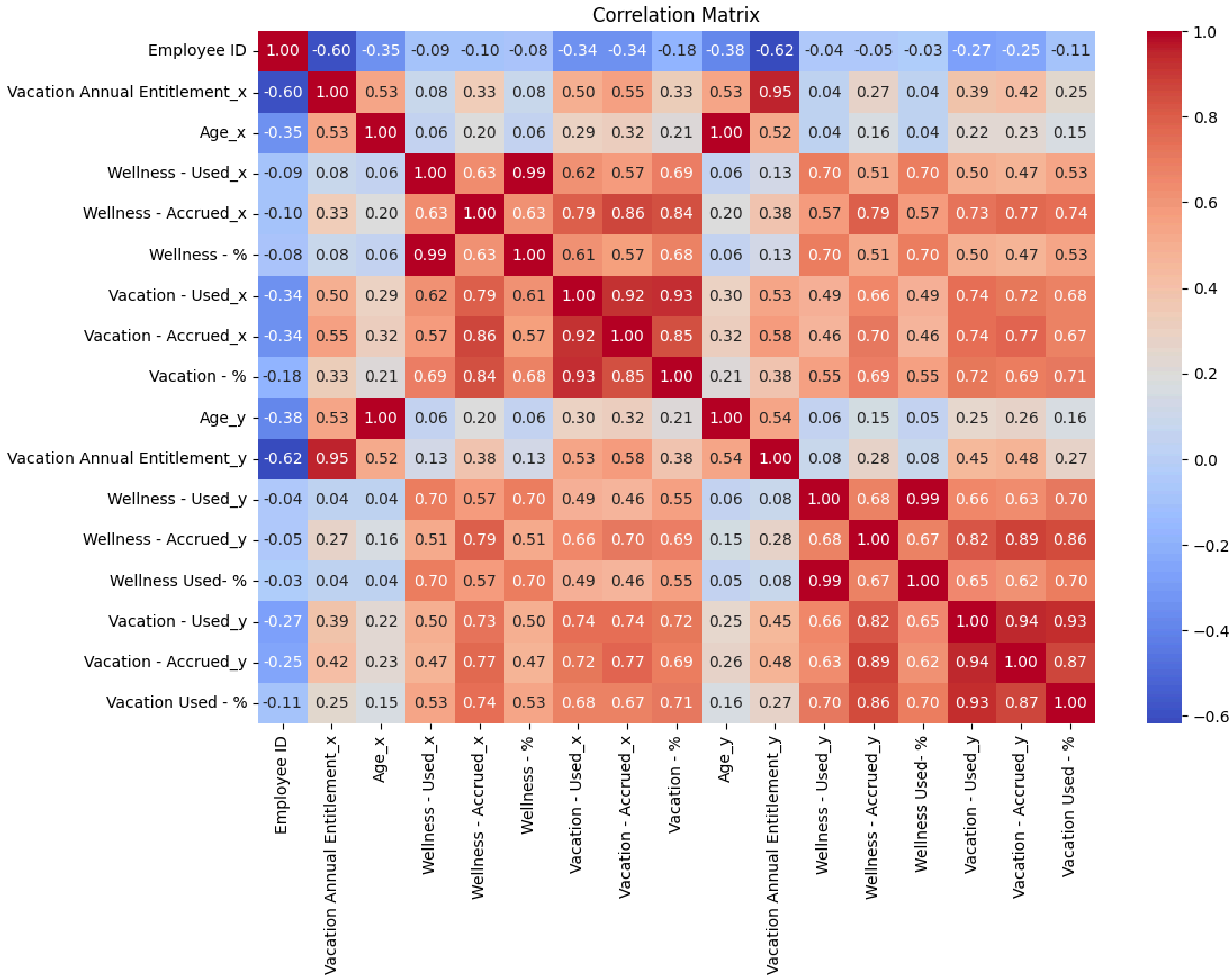
CORRELATION MATRIX

In [8]:
```
# Get numeric columns for correlation matrix
numeric_columns = merged_df.select_dtypes(include=['number']).columns
```

In [9]:
```
# Calculate correlation matrix for numeric columns only
corr_matrix = merged_df[numeric_columns].corr()
```

In [10]:
```
# Plot correlation matrix as a heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f")
```

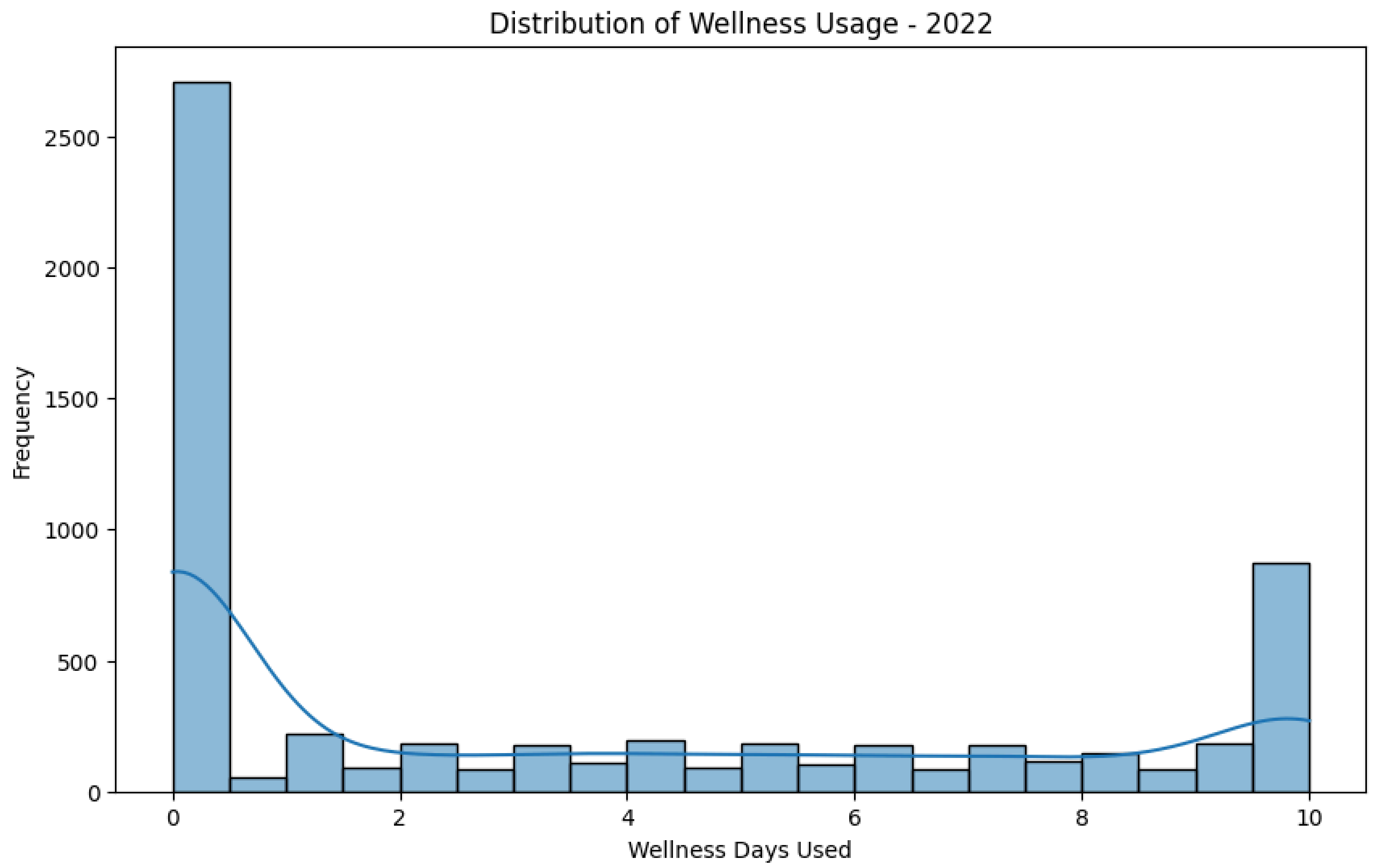
```python
plt.title('Correlation Matrix')
plt.show()
```



DISTRIBUTION OF WELLNESS USAGE

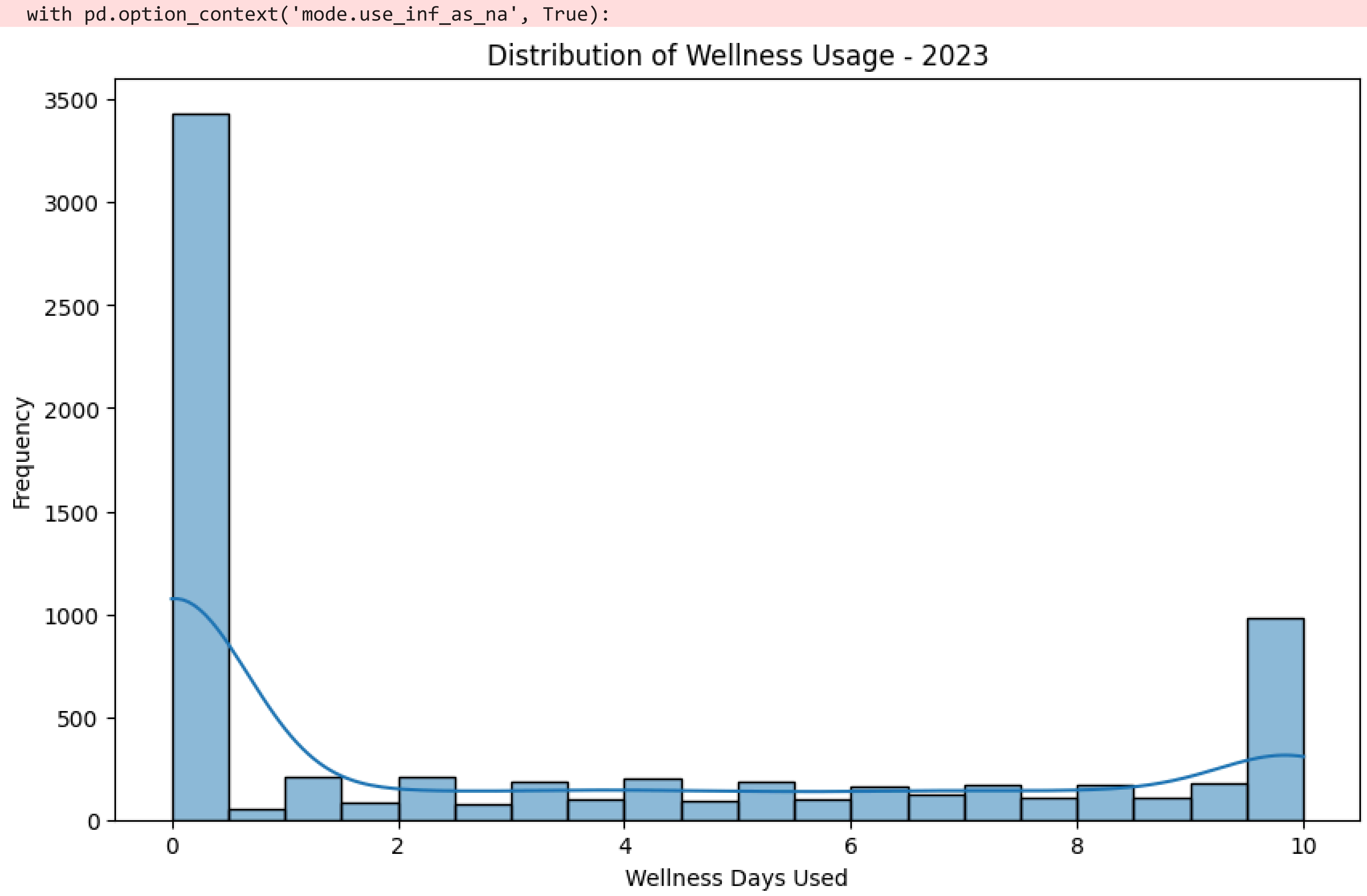
In [11]:
```python
import matplotlib.pyplot as plt
import seaborn as sns

# Plot distribution of wellness usage for 2022
plt.figure(figsize=(10, 6))
sns.histplot(merged_df['Wellness - Used_x'], bins=20, kde=True)
plt.title('Distribution of Wellness Usage - 2022')
plt.xlabel('Wellness Days Used')
plt.ylabel('Frequency')
plt.show()
```

```
/opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



In [12]:
```
# Plot distribution of wellness usage for 2023
plt.figure(figsize=(10, 6))
sns.histplot(merged_df['Wellness - Used_y'], bins=20, kde=True)
plt.title('Distribution of Wellness Usage - 2023')
plt.xlabel('Wellness Days Used')
plt.ylabel('Frequency')
plt.show()
```

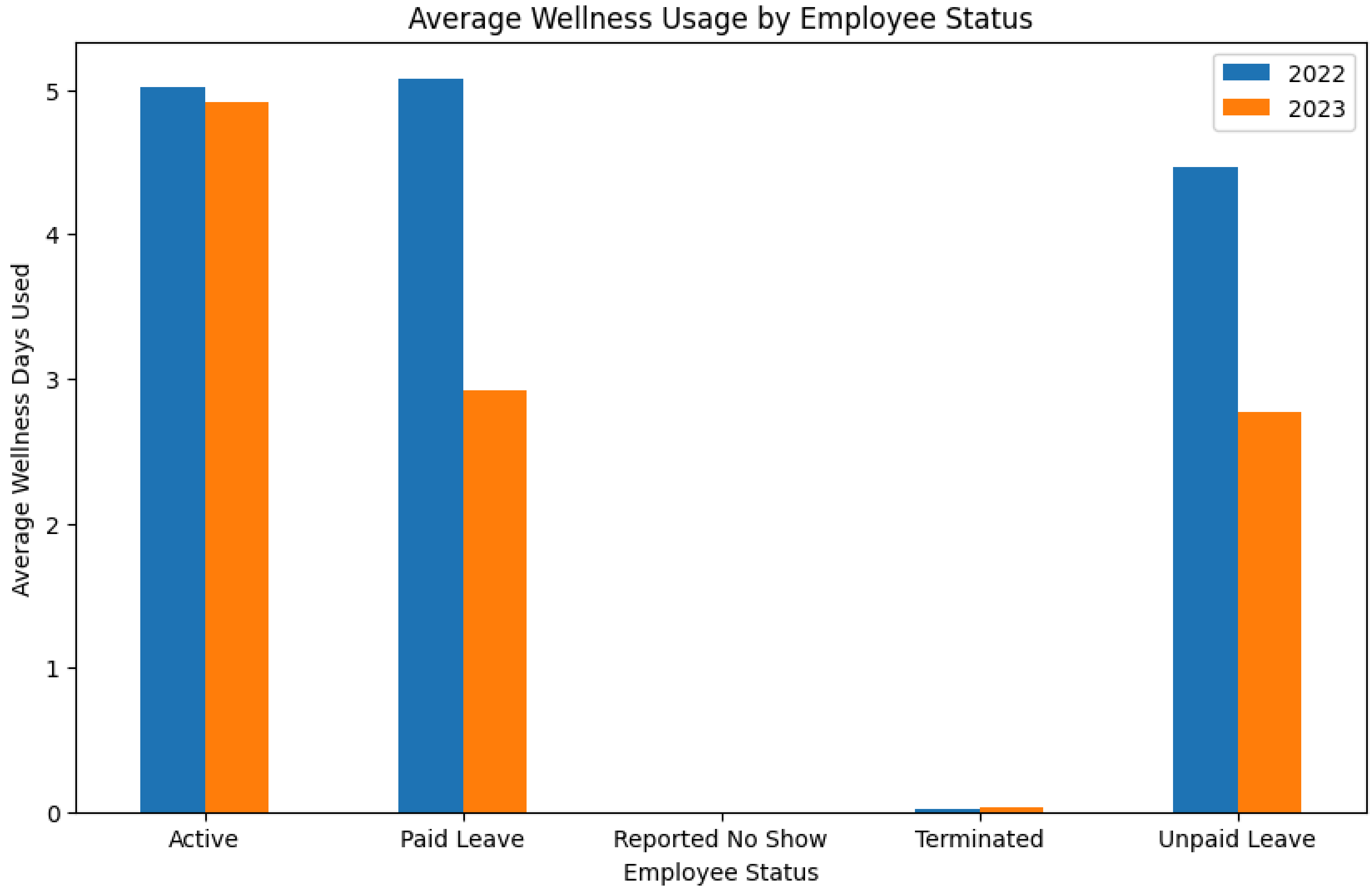
```
/opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and
will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



AVERAGE WELLNESS USED BY EMPLOYEE STATUS

In [13]:
```
# Group data by Employee Status and calculate average wellness usage for both years
wellness_grouped = merged_df.groupby('Employee Status_x')[['Wellness - Used_x', 'Wellness - Used_y']].mean()

# Plot average wellness usage by Employee Status
wellness_grouped.plot(kind='bar', figsize=(10, 6))
plt.title('Average Wellness Usage by Employee Status')
plt.xlabel('Employee Status')
plt.ylabel('Average Wellness Days Used')
plt.xticks(rotation=0)
```

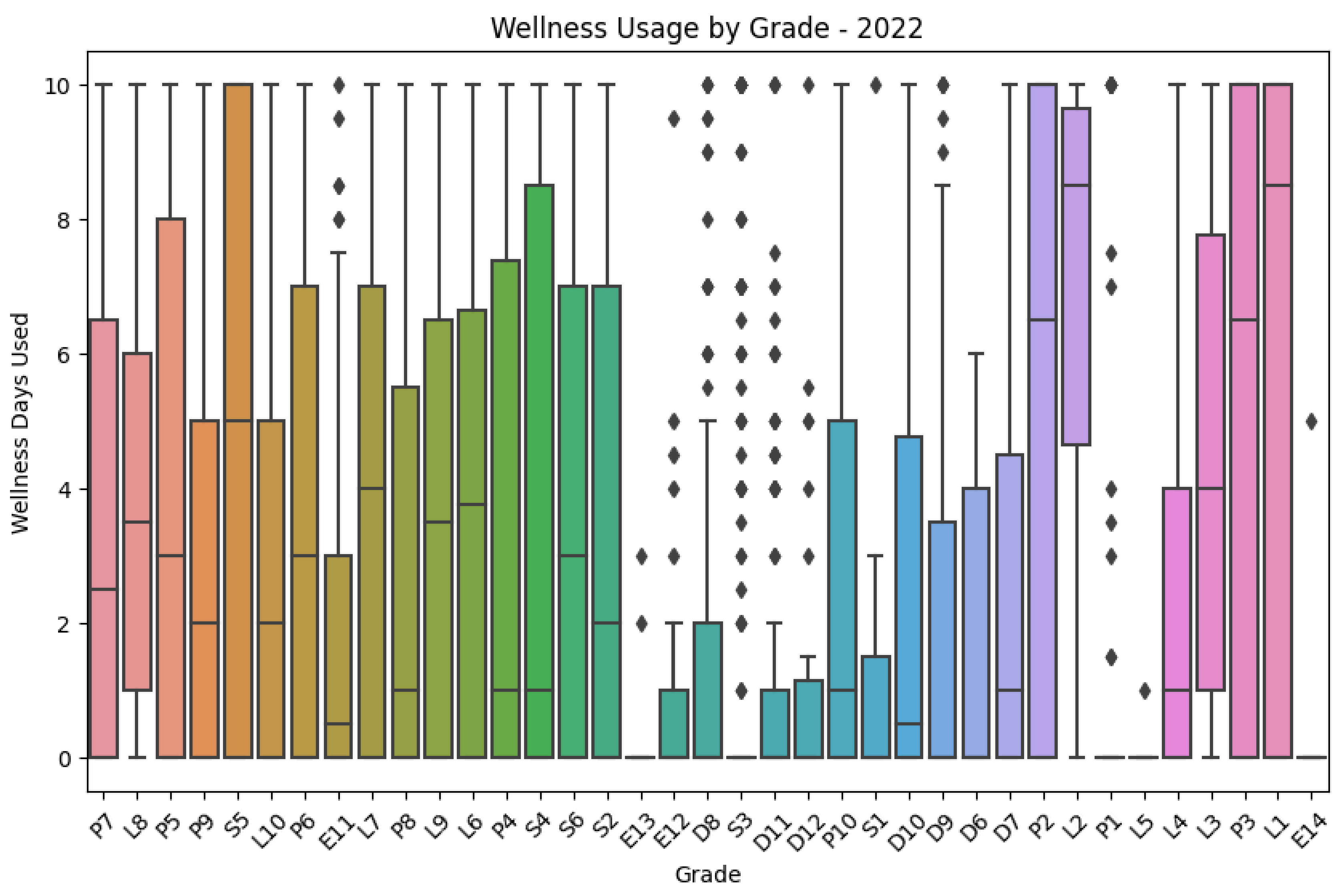
```
plt.legend(['2022', '2023'])
plt.show()
```



WELLNESS USEAGE BY GRADE

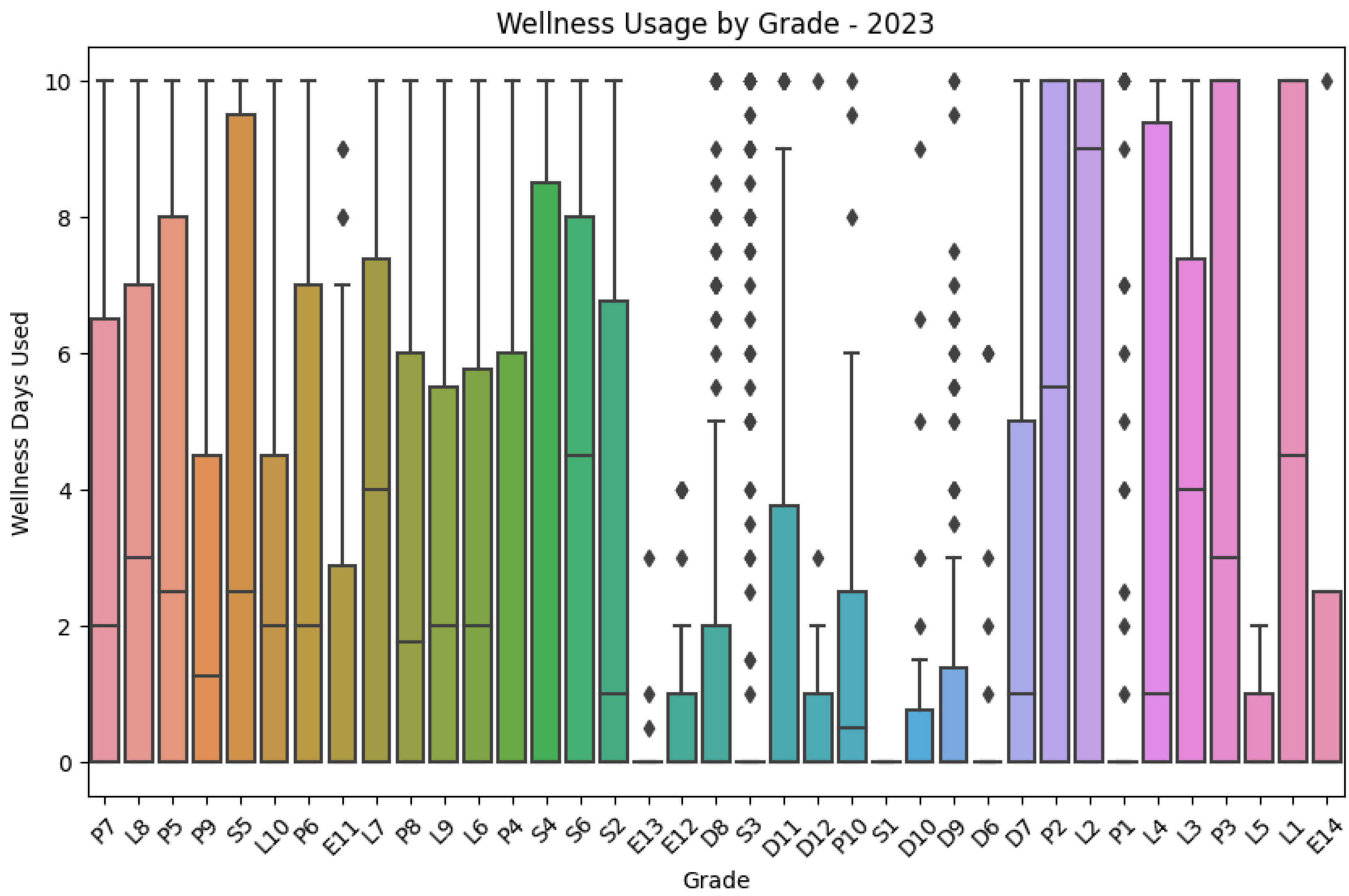
In [14]:

```
# Plot box plot of wellness usage by Grade for 2022
plt.figure(figsize=(10, 6))
sns.boxplot(x='Grade_x', y='Wellness - Used_x', data=merged_df)
plt.title('Wellness Usage by Grade - 2022')
plt.xlabel('Grade')
plt.ylabel('Wellness Days Used')
plt.xticks(rotation=45)
plt.show()
```



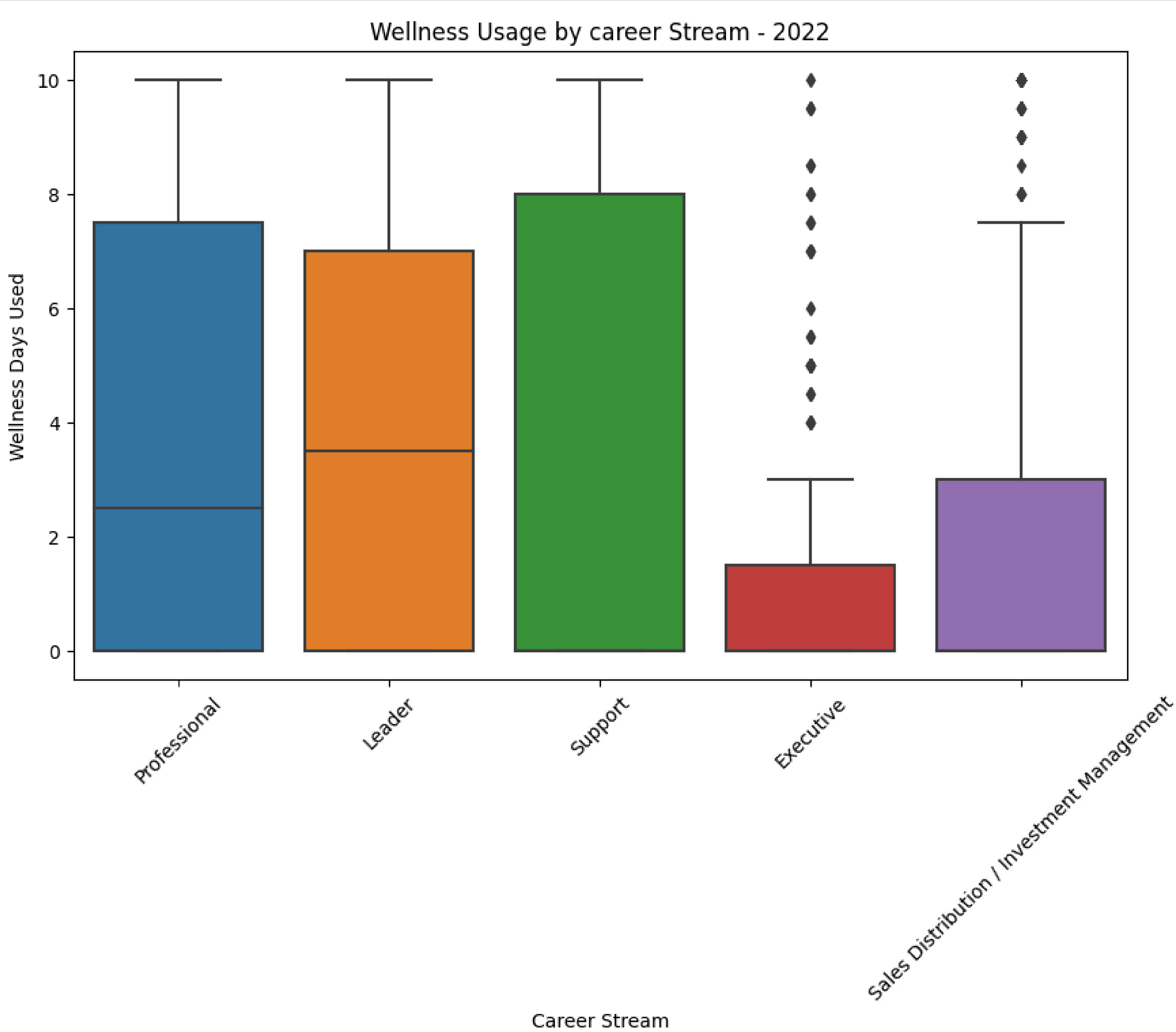
In [15]:

```
# Plot box plot of wellness usage by Grade for 2023
plt.figure(figsize=(10, 6))
sns.boxplot(x='Grade_y', y='Wellness - Used_y', data=merged_df)
plt.title('Wellness Usage by Grade - 2023')
plt.xlabel('Grade')
plt.ylabel('Wellness Days Used')
plt.xticks(rotation=45)
plt.show()
```

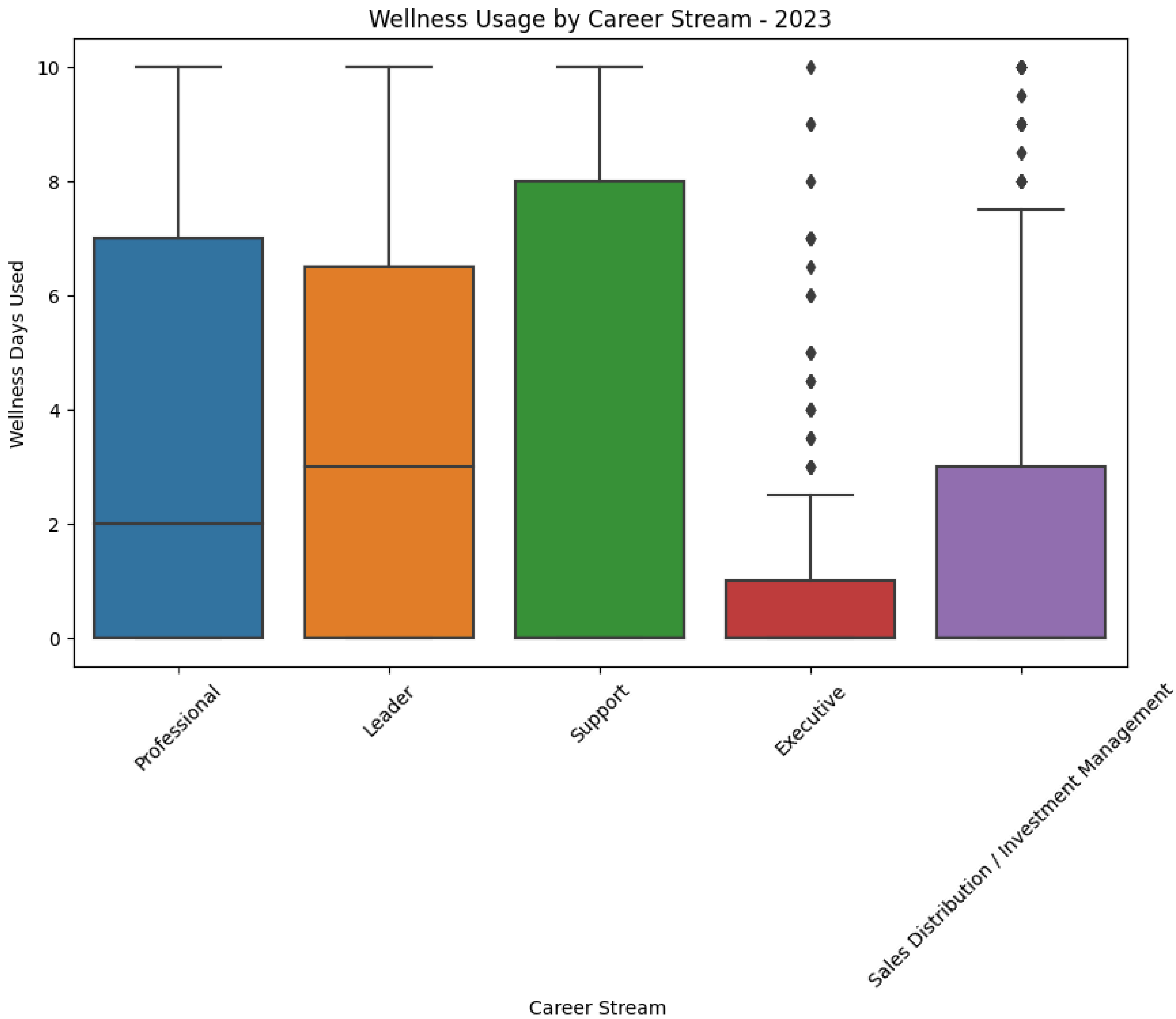


WELLNESS USAGE BY CAREER STREAM

In [16]:
```
# Plot box plot of wellness usage by Career stream for 2022
plt.figure(figsize=(10, 6))
sns.boxplot(x='Career Stream_x', y='Wellness - Used_x', data=merged_df)
plt.title('Wellness Usage by career Stream - 2022')
plt.xlabel('Career Stream')
plt.ylabel('Wellness Days Used')
plt.xticks(rotation=45)
plt.show()
```

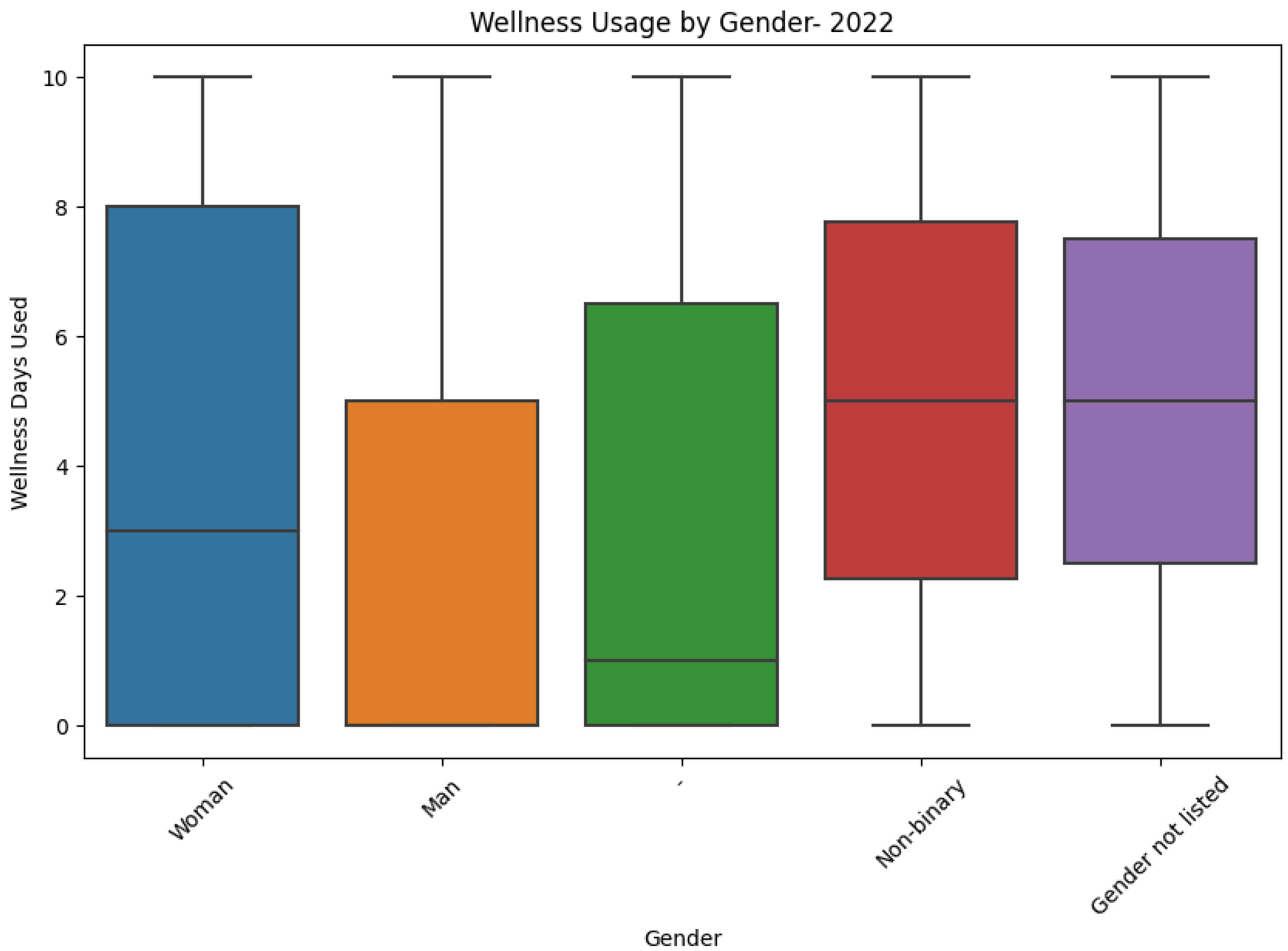


In [17]:
```
# Plot box plot of wellness usage by Career stream  for 2023
plt.figure(figsize=(10, 6))
sns.boxplot(x='Career Stream_y', y='Wellness - Used_y', data=merged_df)
plt.title('Wellness Usage by Career Stream - 2023')
plt.xlabel('Career Stream ')
plt.ylabel('Wellness Days Used')
plt.xticks(rotation=45)
plt.show()
```

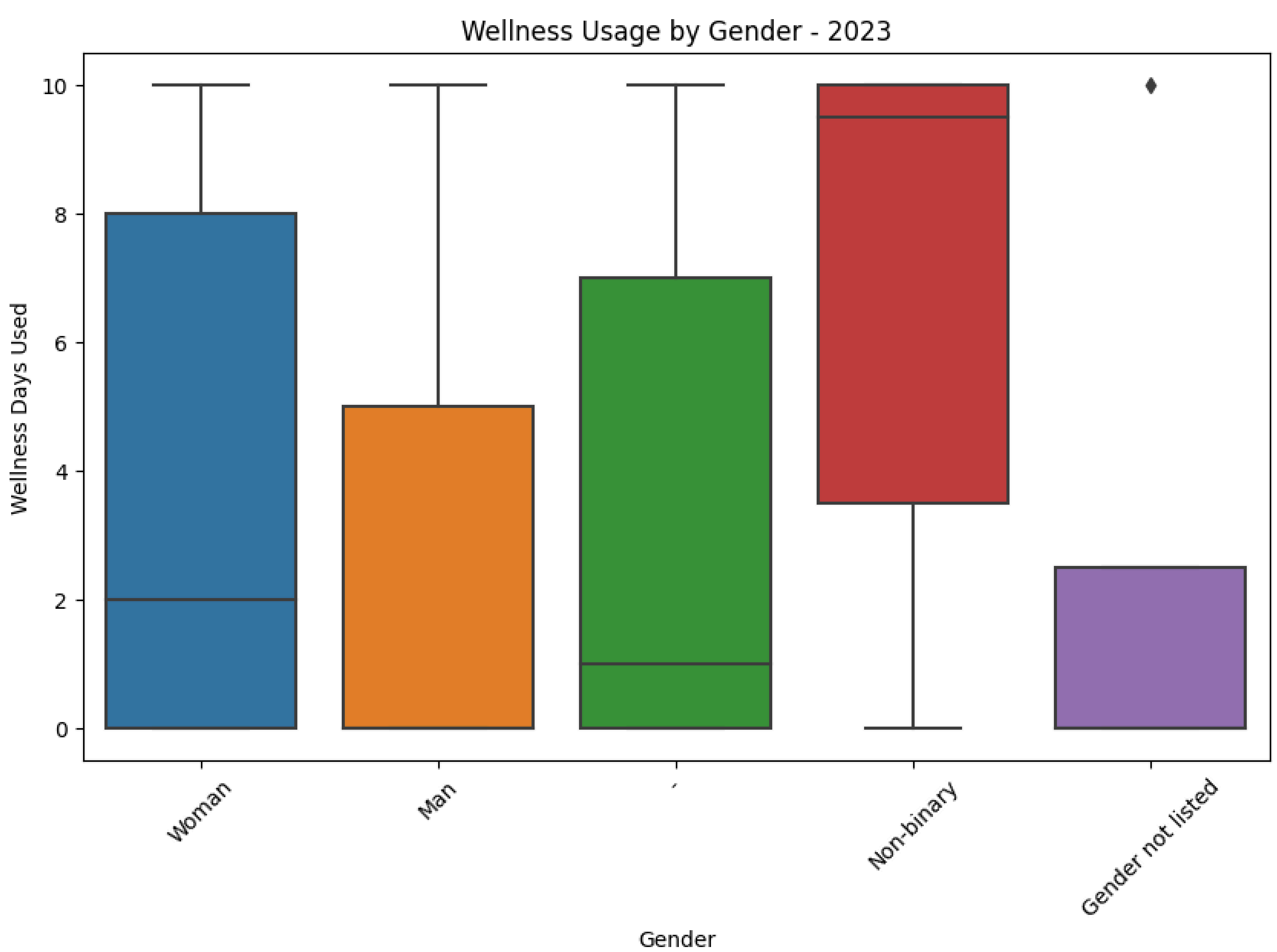


WELLNESS USAGE BY GENDER

In [18]:
```
# Plot box plot of wellness usage by Gender for 2022
plt.figure(figsize=(10, 6))
sns.boxplot(x='Gender_x', y='Wellness - Used_x', data=merged_df)
plt.title('Wellness Usage by Gender- 2022')
plt.xlabel('Gender')
plt.ylabel('Wellness Days Used')
plt.xticks(rotation=45)
plt.show()
```

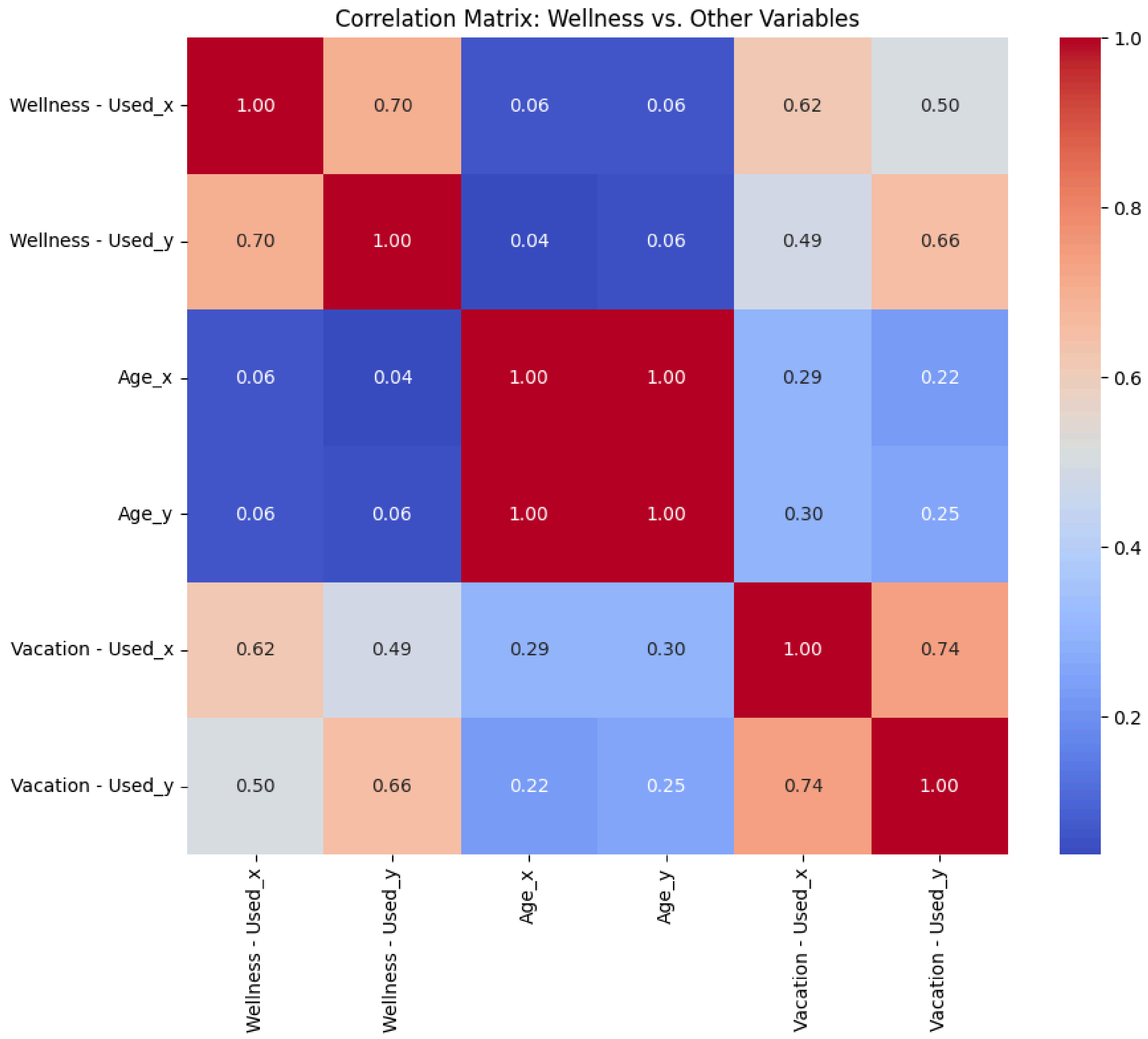


In [19]:
```
# Plot box plot of wellness usage by Gender for 2023
plt.figure(figsize=(10, 6))
sns.boxplot(x='Gender_y', y='Wellness - Used_y', data=merged_df)
plt.title('Wellness Usage by Gender - 2023')
plt.xlabel('Gender')
plt.ylabel('Wellness Days Used')
plt.xticks(rotation=45)
plt.show()
```



CORRELATION BETWEEN WELLNESS USED, AGE, AND VACATION USED

In [20]:
```
# Calculate correlations between wellness usage and other numeric variables
wellness_corr = merged_df[['Wellness - Used_x', 'Wellness - Used_y', 'Age_x','Age_y', 'Vacation - Used_x', 'Vacation - U

# Plot correlation heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(wellness_corr, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix: Wellness vs. Other Variables')
plt.show()
```



Hypothesis Testing: Wellness Usage and Employee Retention

Hypothesis

- **Null Hypothesis (H0):** There is no significant difference in wellness usage between employees who stay and employees who leave the company.
- **Alternative Hypothesis (H1):** Employees who stay use significantly more wellness benefits compared to employees who leave.

Methodology

- I performed a two-sample t-test to compare the mean wellness usage between employees who are active (stay) and those who are terminated (leave).
- The significance level (alpha) chosen was 0.05.

In [21]:
```
import scipy.stats as stats

# Separate wellness usage data for employees who stay and those who leave
wellness_stay = merged_df.loc[merged_df['Employee Status_x'] == 'Active', 'Wellness - Used_x']
wellness_leave = merged_df.loc[merged_df['Employee Status_x'] == 'Terminated', 'Wellness - Used_x']
```

In [22]:
```
# Perform two-sample t-test
t_stat, p_value = stats.ttest_ind(wellness_stay, wellness_leave, equal_var=True)
```

In [23]:
```
# Print t-statistic and p-value
print("T-statistic:", t_stat)
print("P-value:", p_value)
```

```
T-statistic: 59.54929608933553
P-value: 0.0
```

In [24]:
```
# Interpret the results
alpha = 0.05  # significance level
if p_value < alpha:
    print("Reject null hypothesis: There is a significant difference in wellness usage between employees who stay and th
else:
    print("Fail to reject null hypothesis: There is no significant difference in wellness usage between employees who st
```

```
Reject null hypothesis: There is a significant difference in wellness usage between employees who stay and those who le
ave.
```

Results

- **T-Statistic:** 59.55
- **P-Value:** 0.0 (rounded)

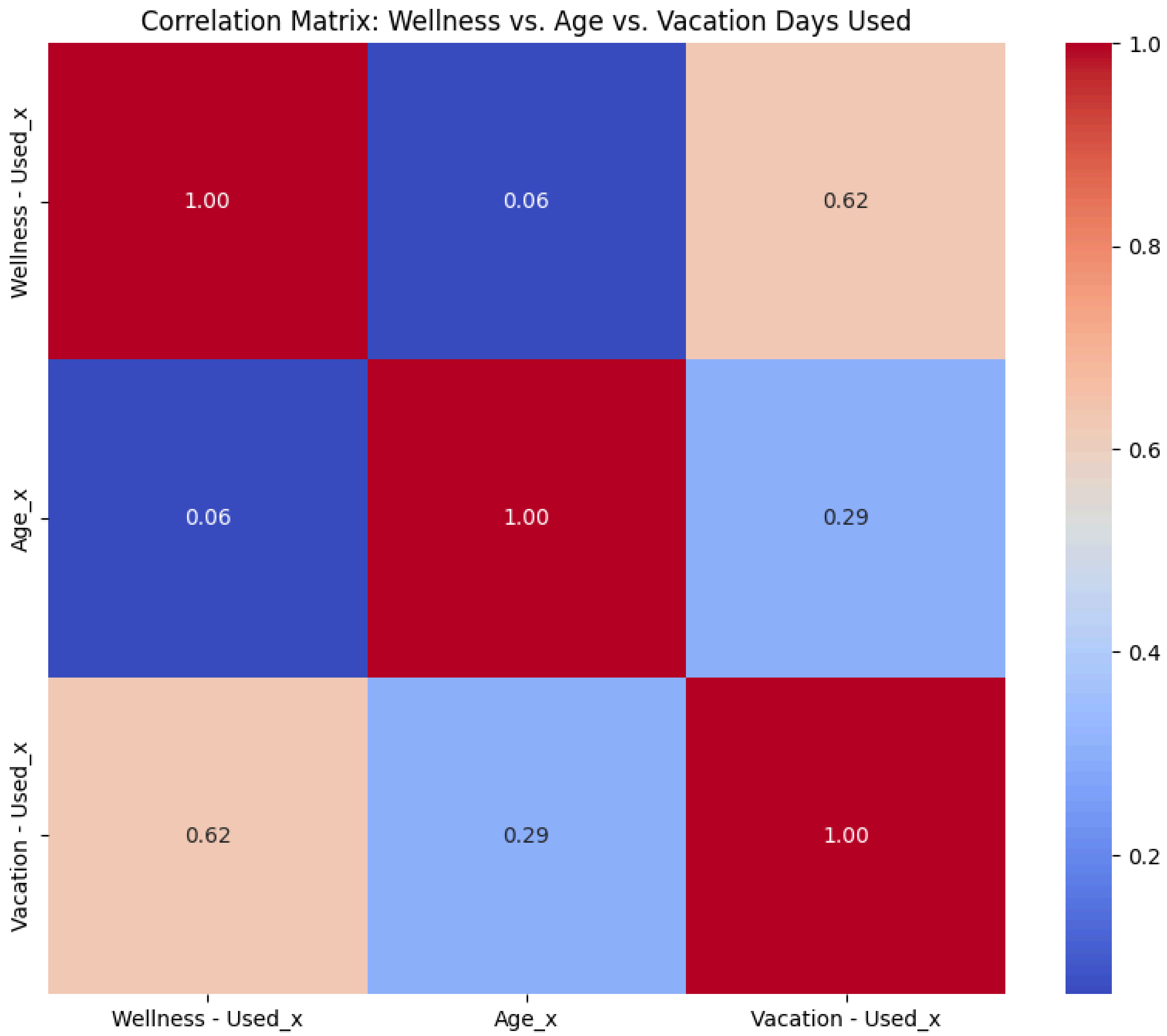
Conclusion

- Based on the results of the t-test:
    - **Reject Null Hypothesis:** There is a significant difference in wellness usage between employees who stay and those who leave the company.
    - Employees who stay tend to use significantly more wellness benefits compared to employees who leave.

Correlation Analysis:

I'll start by calculating the correlation coefficients between wellness usage and other relevant variables such as age, vacation days used, etc. This will help us understand the strength and direction of these relationships.

In [25]:
```
# Calculate correlation matrix
correlation_matrix = merged_df[['Wellness - Used_x', 'Age_x', 'Vacation - Used_x']].corr()

# Plot correlation heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix: Wellness vs. Age vs. Vacation Days Used')
plt.show()

#Print correlation matric
print("Correlation Matrix:")
print(corr_matrix)
```



```
Correlation Matrix:
                              Employee ID  Vacation Annual Entitlement_x  \
Employee ID                      1.000000                      -0.602802   
Vacation Annual Entitlement_x   -0.602802                       1.000000   
Age_x                           -0.350633                       0.534915   
Wellness - Used_x               -0.086591                       0.082760   
Wellness - Accrued_x            -0.099079                       0.332575   
Wellness - %                    -0.079062                       0.077978   
Vacation - Used_x               -0.344016                       0.502085   
Vacation - Accrued_x            -0.336177                       0.553775   
Vacation - %                    -0.180767                       0.329327   
Age_y                           -0.378433                       0.534970   
Vacation Annual Entitlement_y   -0.617874                       0.953540   
Wellness - Used_y               -0.041682                       0.035086   
Wellness - Accrued_y            -0.049798                       0.269746   
Wellness Used- %                -0.033185                       0.035113   
Vacation - Used_y               -0.265518                       0.387168   
Vacation - Accrued_y            -0.245354                       0.417957   
Vacation Used - %               -0.105911                       0.247412   

                                  Age_x  Wellness - Used_x  \
Employee ID                   -0.350633          -0.086591   
Vacation Annual Entitlement_x  0.534915           0.082760   
Age_x                          1.000000           0.062610   
Wellness - Used_x              0.062610           1.000000   
Wellness - Accrued_x           0.196006           0.631080   
Wellness - %                   0.061794           0.994040   
Vacation - Used_x              0.294099           0.620251   
Vacation - Accrued_x           0.316252           0.572308   
Vacation - %                   0.207012           0.685026   
Age_y                          0.999999           0.064155   
Vacation Annual Entitlement_y  0.519121           0.132169   
Wellness - Used_y              0.036885           0.697891   
Wellness - Accrued_y           0.160822           0.511708   
Wellness Used- %               0.036685           0.697990   
Vacation - Used_y              0.221896           0.500125   
Vacation - Accrued_y           0.232501           0.471604   
Vacation Used - %              0.153658           0.532483   

                              Wellness - Accrued_x  Wellness - %  \
Employee ID                              -0.099079     -0.079062   
Vacation Annual Entitlement_x             0.332575      0.077978   
Age_x                                     0.196006      0.061794   
Wellness - Used_x                         0.631080      0.994040   
Wellness - Accrued_x                      1.000000      0.626796   
Wellness - %                              0.626796      1.000000   
Vacation - Used_x                         0.789010      0.612503   
Vacation - Accrued_x                      0.864607      0.565545   
Vacation - %                              0.841832      0.677569   
Age_y                                     0.195404      0.063338   
Vacation Annual Entitlement_y             0.381552      0.127193   
Wellness - Used_y                         0.572411      0.699353   
Wellness - Accrued_y                      0.786154      0.513710   
Wellness Used- %                          0.572475      0.699450   
Vacation - Used_y                         0.729010      0.499262   
Vacation - Accrued_y                      0.767673      0.471552   
Vacation Used - %                         0.737329      0.533616   

                              Vacation - Used_x  Vacation - Accrued_x  \
Employee ID                           -0.344016             -0.336177   
Vacation Annual Entitlement_x          0.502085              0.553775   
Age_x                                  0.294099              0.316252   
Wellness - Used_x                      0.620251              0.572308   
Wellness - Accrued_x                   0.789010              0.864607   
Wellness - %                           0.612503              0.565545   
Vacation - Used_x                      1.000000              0.921932   
Vacation - Accrued_x                   0.921932              1.000000   
Vacation - %                           0.930321              0.847012   
Age_y                                  0.295451              0.316811   
Vacation Annual Entitlement_y          0.530218              0.584768   
Wellness - Used_y                      0.485306              0.457552   
Wellness - Accrued_y                   0.655662              0.704196   
Wellness Used- %                       0.485380              0.457631   
Vacation - Used_y                      0.740302              0.737149   
Vacation - Accrued_y                   0.715766              0.773807   
Vacation Used - %                      0.675164              0.672772   

                              Vacation - %     Age_y  \
Employee ID                      -0.180767 -0.378433   
Vacation Annual Entitlement_x     0.329327  0.534970   
Age_x                             0.207012  0.999999   
Wellness - Used_x                 0.685026  0.064155   
Wellness - Accrued_x              0.841832  0.195404   
Wellness - %                      0.677569  0.063338   
Vacation - Used_x                 0.930321  0.295451   
Vacation - Accrued_x              0.847012  0.316811   
Vacation - %                      1.000000  0.209061   
Age_y                             0.209061  1.000000   
Vacation Annual Entitlement_y     0.376393  0.535061   
Wellness - Used_y                 0.552744  0.055416   
Wellness - Accrued_y              0.688758  0.152710   
Wellness Used- %                  0.552821  0.054413   
Vacation - Used_y                 0.717371  0.249812   
Vacation - Accrued_y              0.691663  0.260653   
Vacation Used - %                 0.706062  0.157626   
```

```
                              Vacation Annual Entitlement_y  \
Employee ID                                       -0.617874
Vacation Annual Entitlement_x                      0.953540
Age_x                                              0.519121
Wellness - Used_x                                  0.132169
Wellness - Accrued_x                               0.381552
Wellness - %                                       0.127193
Vacation - Used_x                                  0.530218
Vacation - Accrued_x                               0.584768
Vacation - %                                       0.376393
Age_y                                              0.535061
Vacation Annual Entitlement_y                      1.000000
Wellness - Used_y                                  0.083247
Wellness - Accrued_y                               0.277843
Wellness Used- %                                   0.077647
Vacation - Used_y                                  0.446862
Vacation - Accrued_y                               0.476740
Vacation Used - %                                  0.270857

                              Wellness - Used_y  Wellness - Accrued_y  \
Employee ID                           -0.041682             -0.049798
Vacation Annual Entitlement_x          0.035086              0.269746
Age_x                                  0.036885              0.160822
Wellness - Used_x                      0.697891              0.511708
Wellness - Accrued_x                   0.572411              0.786154
Wellness - %                           0.699353              0.513710
Vacation - Used_x                      0.485306              0.655662
Vacation - Accrued_x                   0.457552              0.704196
Vacation - %                           0.552744              0.688758
Age_y                                  0.055416              0.152710
Vacation Annual Entitlement_y          0.083247              0.277843
Wellness - Used_y                      1.000000              0.681465
Wellness - Accrued_y                   0.681465              1.000000
Wellness Used- %                       0.994161              0.673699
Vacation - Used_y                      0.656795              0.822174
Vacation - Accrued_y                   0.631674              0.885636
Vacation Used - %                      0.703030              0.859778

                              Wellness Used- %  Vacation - Used_y  \
Employee ID                          -0.033185          -0.265518
Vacation Annual Entitlement_x         0.035113           0.387168
Age_x                                 0.036685           0.221896
Wellness - Used_x                     0.697990           0.500125
Wellness - Accrued_x                  0.572475           0.729010
Wellness - %                          0.699450           0.499262
Vacation - Used_x                     0.485380           0.740302
Vacation - Accrued_x                  0.457631           0.737149
Vacation - %                          0.552821           0.717371
Age_y                                 0.054413           0.249812
Vacation Annual Entitlement_y         0.077647           0.446862
Wellness - Used_y                     0.994161           0.656795
Wellness - Accrued_y                  0.673699           0.822174
Wellness Used- %                      1.000000           0.648686
Vacation - Used_y                     0.648686           1.000000
Vacation - Accrued_y                  0.624710           0.941791
Vacation Used - %                     0.699995           0.933705

                              Vacation - Accrued_y  Vacation Used - %
Employee ID                              -0.245354          -0.105911
Vacation Annual Entitlement_x             0.417957           0.247412
Age_x                                     0.232501           0.153658
Wellness - Used_x                         0.471604           0.532483
Wellness - Accrued_x                      0.767673           0.737329
Wellness - %                              0.471552           0.533616
Vacation - Used_x                         0.715766           0.675164
Vacation - Accrued_x                      0.773807           0.672772
Vacation - %                              0.691663           0.706062
Age_y                                     0.260653           0.157626
Vacation Annual Entitlement_y             0.476740           0.270857
Wellness - Used_y                         0.631674           0.703030
Wellness - Accrued_y                      0.885636           0.859778
Wellness Used- %                          0.624710           0.699995
Vacation - Used_y                         0.941791           0.933705
Vacation - Accrued_y                      1.000000           0.870477
Vacation Used - %                         0.870477           1.000000
```

Regression Analysis:

I perform a regression analysis to model the relationship between wellness usage (independent variable) and employee retention (dependent variable). Since employee retention is a binary outcome (stay or leave), I use logistic regression.

In [26]:
```
import pandas as pd
import statsmodels.api as sm

# Drop rows with missing values in relevant columns
merged_df_cleaned = merged_df.dropna(subset=['Wellness - Used_x', 'Employee Status_x'])

# Define predictors (X) and target (y) after handling missing values
X = merged_df_cleaned[['Wellness - Used_x']]
y = (merged_df_cleaned['Employee Status_x'] == 'Active').astype(int)  # Convert 'Active' to 1, 'Terminated' to 0
```

```python
# Add constant to X for intercept in regression
X = sm.add_constant(X)

# Fit logistic regression model
logit_model = sm.Logit(y, X)
result = logit_model.fit()

# Print model summary
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.419404
         Iterations 8
                           Logit Regression Results
==============================================================================
Dep. Variable:      Employee Status_x   No. Observations:                 6044
Model:                          Logit   Df Residuals:                     6042
Method:                           MLE   Df Model:                            1
Date:                Thu, 13 Jun 2024   Pseudo R-squ.:                  0.3533
Time:                        11:27:33   Log-Likelihood:                -2534.9
converged:                       True   LL-Null:                       -3919.7
Covariance Type:            nonrobust   LLR p-value:                     0.000
=====================================================================================
                        coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------
const                -0.6206      0.038    -16.327      0.000      -0.695      -0.546
Wellness - Used_x     0.6801      0.023     30.121      0.000       0.636       0.724
=====================================================================================
```

SURVIVAL ANALYSIS **

setting Survival Analysis: Time to Employee Termination

Approach

I conducted survival analysis to understand the duration until an employee terminates employment using the Kaplan-Meier estimator.

Methodology

1. **Data Preparation:**
   - I calculated the tenure (time from hire to termination) in days using the 'Hire Date' and 'Termination Date' columns.
   - Rows with missing tenure values were dropped to ensure data integrity.
2. **Survival Analysis:**
   - I used the KaplanMeierFitter from the lifelines library to estimate the survival function.
   - The analysis was based on time-to-event data (tenure) and event status (termination).
3. **Visualization:**
   - A Kaplan-Meier survival curve was plotted to visualize the probability of employees not leaving the company over time.

Results

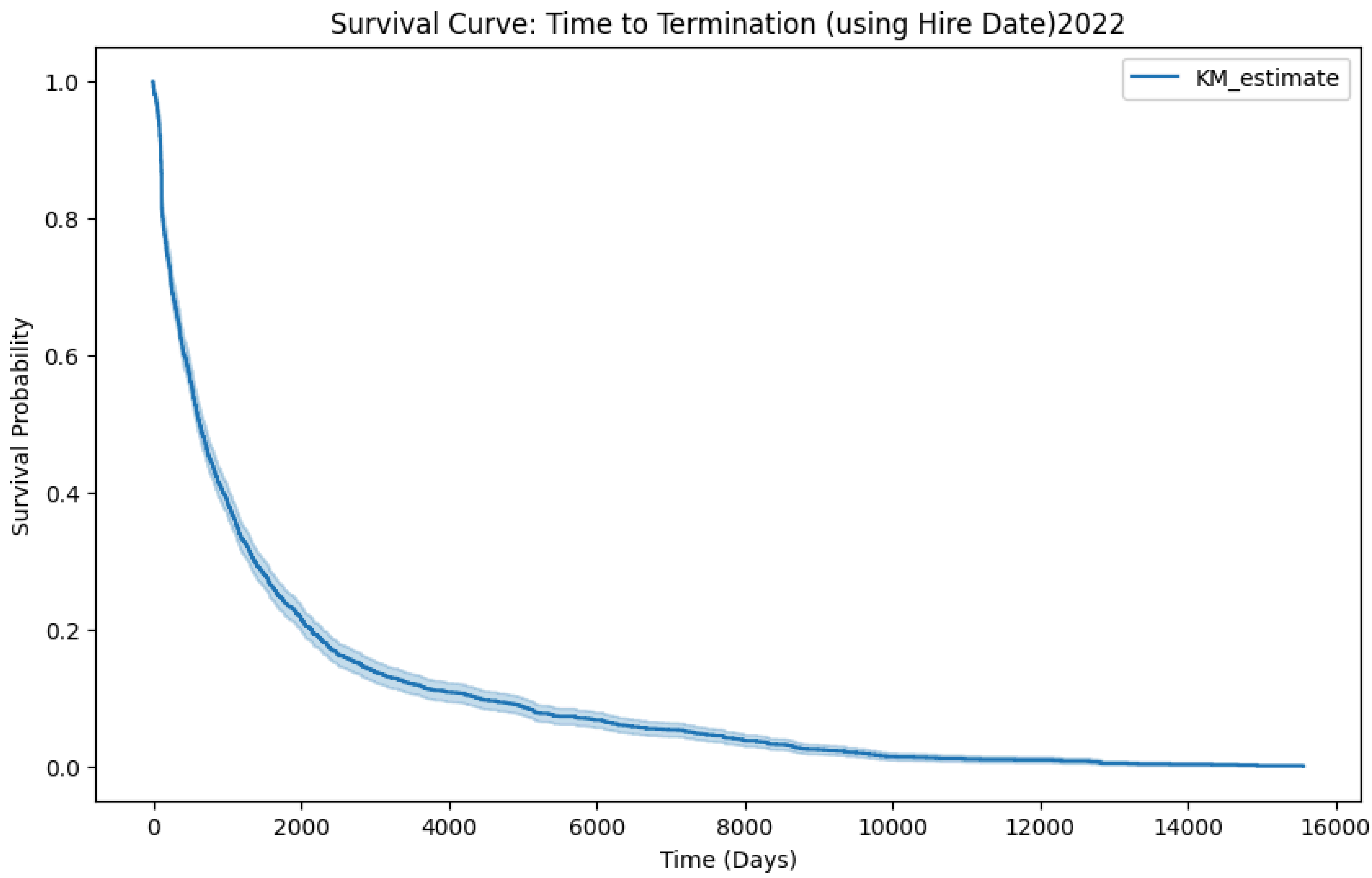
The survival curve provides insights into the duration employees stay with the company before termination. It helps in understanding the attrition pattern and can guide HR strategies for employee retention and management.

In [27]:
```
!pip install lifelines
```

```
Collecting lifelines
  Downloading lifelines-0.28.0-py3-none-any.whl.metadata (3.2 kB)
Requirement already satisfied: numpy<2.0,>=1.14.0 in /opt/conda/lib/python3.10/site-packages (from lifelines) (1.26.4)
Requirement already satisfied: scipy>=1.2.0 in /opt/conda/lib/python3.10/site-packages (from lifelines) (1.11.4)
Requirement already satisfied: pandas>=1.2.0 in /opt/conda/lib/python3.10/site-packages (from lifelines) (2.2.2)
Requirement already satisfied: matplotlib>=3.0 in /opt/conda/lib/python3.10/site-packages (from lifelines) (3.7.5)
Collecting autograd>=1.5 (from lifelines)
  Downloading autograd-1.6.2-py3-none-any.whl.metadata (706 bytes)
Collecting autograd-gamma>=0.3 (from lifelines)
  Downloading autograd-gamma-0.5.0.tar.gz (4.0 kB)
  Preparing metadata (setup.py) ... done
Collecting formulaic>=0.2.2 (from lifelines)
  Downloading formulaic-1.0.1-py3-none-any.whl.metadata (6.1 kB)
Requirement already satisfied: future>=0.15.2 in /opt/conda/lib/python3.10/site-packages (from autograd>=1.5->lifeline
s) (1.0.0)
Collecting interface-meta>=1.2.0 (from formulaic>=0.2.2->lifelines)
  Downloading interface_meta-1.3.0-py3-none-any.whl.metadata (6.7 kB)
Requirement already satisfied: typing-extensions>=4.2.0 in /opt/conda/lib/python3.10/site-packages (from formulaic>=0.
2.2->lifelines) (4.9.0)
Requirement already satisfied: wrapt>=1.0 in /opt/conda/lib/python3.10/site-packages (from formulaic>=0.2.2->lifelines)
(1.14.1)
Requirement already satisfied: contourpy>=1.0.1 in /opt/conda/lib/python3.10/site-packages (from matplotlib>=3.0->lifel
ines) (1.2.0)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.10/site-packages (from matplotlib>=3.0->lifeline
s) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /opt/conda/lib/python3.10/site-packages (from matplotlib>=3.0->life
lines) (4.47.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /opt/conda/lib/python3.10/site-packages (from matplotlib>=3.0->life
lines) (1.4.5)
Requirement already satisfied: packaging>=20.0 in /opt/conda/lib/python3.10/site-packages (from matplotlib>=3.0->lifeli
nes) (21.3)
Requirement already satisfied: pillow>=6.2.0 in /opt/conda/lib/python3.10/site-packages (from matplotlib>=3.0->lifeline
s) (9.5.0)
Requirement already satisfied: pyparsing>=2.3.1 in /opt/conda/lib/python3.10/site-packages (from matplotlib>=3.0->lifel
ines) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in /opt/conda/lib/python3.10/site-packages (from matplotlib>=3.0->l
ifelines) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.10/site-packages (from pandas>=1.2.0->lifelines)
(2023.3.post1)
Requirement already satisfied: tzdata>=2022.7 in /opt/conda/lib/python3.10/site-packages (from pandas>=1.2.0->lifeline
s) (2023.4)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.10/site-packages (from python-dateutil>=2.7->matplotl
ib>=3.0->lifelines) (1.16.0)
Downloading lifelines-0.28.0-py3-none-any.whl (349 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 349.2/349.2 kB 10.3 MB/s eta 0:00:00
Downloading autograd-1.6.2-py3-none-any.whl (49 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 49.3/49.3 kB 2.5 MB/s eta 0:00:00
Downloading formulaic-1.0.1-py3-none-any.whl (94 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 94.2/94.2 kB 4.1 MB/s eta 0:00:00
Downloading interface_meta-1.3.0-py3-none-any.whl (14 kB)
Building wheels for collected packages: autograd-gamma
  Building wheel for autograd-gamma (setup.py) ... done
  Created wheel for autograd-gamma: filename=autograd_gamma-0.5.0-py3-none-any.whl size=4031 sha256=319178f2c2339f6a100
cfd5d8fcb55c36bd961a84085de1dfc6b593157049e58
  Stored in directory: /root/.cache/pip/wheels/25/cc/e0/ef2969164144c899fedb22b338f6703e2b9cf46eeebf254991
Successfully built autograd-gamma
Installing collected packages: interface-meta, autograd, autograd-gamma, formulaic, lifelines
Successfully installed autograd-1.6.2 autograd-gamma-0.5.0 formulaic-1.0.1 interface-meta-1.3.0 lifelines-0.28.0
```

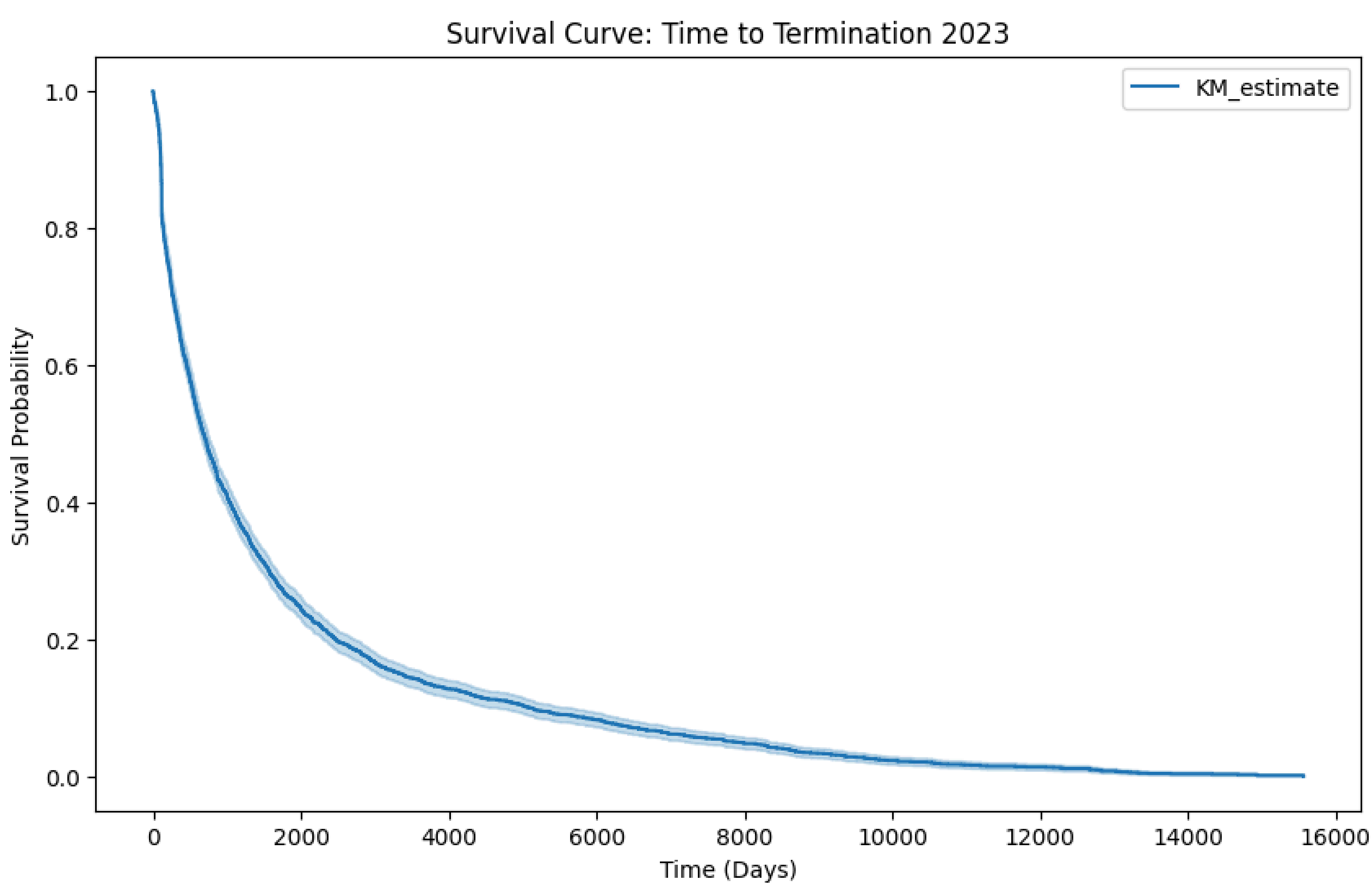
In [28]:

```python
from lifelines import KaplanMeierFitter
import matplotlib.pyplot as plt
import pandas as pd


# Calculate tenure (time until termination) in days
merged_df['Tenure'] = (merged_df['Termination Date_x'] - merged_df['Hire Date_x']).dt.days

# Drop rows with missing tenure values (if any)
merged_df_cleaned = merged_df.dropna(subset=['Tenure', 'Employee Status_x'])

# Create a Kaplan-Meier estimator object
kmf = KaplanMeierFitter()

# Define time-to-event (survival time) and event status (1 for termination, 0 for ongoing employment)
time = merged_df_cleaned['Tenure']
event = (merged_df_cleaned['Employee Status_x'] == 'Terminated').astype(int)

# Fit the Kaplan-Meier estimator
kmf.fit(time, event)

# Plot the survival curve
plt.figure(figsize=(10, 6))
kmf.plot()
plt.title('Survival Curve: Time to Termination (using Hire Date)2022')
plt.xlabel('Time (Days)')
plt.ylabel('Survival Probability')
plt.show()
```



In [29]:

```
# Calculate tenure (time from hire to termination) in days
merged_df['Tenure'] = (merged_df['Termination Date_y'] - merged_df['Hire Date_y']).dt.days

# Drop rows with missing tenure values (if any)
merged_df_cleaned = merged_df.dropna(subset=['Tenure', 'Employee Status_y'])

# Create a Kaplan-Meier estimator object
kmf = KaplanMeierFitter()

# Define time-to-event (survival time) and event status (1 for termination, 0 for ongoing employment)
time = merged_df_cleaned['Tenure']
event = (merged_df_cleaned['Employee Status_y'] == 'Terminated').astype(int)

# Fit the Kaplan-Meier estimator
kmf.fit(time, event)

# Plot the survival curve
plt.figure(figsize=(10, 6))
kmf.plot()
plt.title('Survival Curve: Time to Termination 2023')
plt.xlabel('Time (Days)')
plt.ylabel('Survival Probability')
plt.show()
```



PREDICTIVE MODELING

Predictive Modeling: Random Forest Classifier for Employee Retention

Approach

I capabilities employed a Random Forest Classifier to predict the likelihood of an employee staying or leaving the company based on various features including wellness usage, age, and vacation days used.

Methodology

1. **Data Preparation:**
   - I selected 'Wellness - Used_x', 'Age', and 'Vacation - Used_x' as predictors (X) and 'Employee Status' as the target (y).
2. **Model Training and Testing:**
   - The dataset was split into training and testing sets (80% training, 20% testing) using `train_test_split` from sklearn.
   - I trained the Random Forest Classifier with 100 estimators using `RandomForestClassifier` from sklearn.
3. **Model Evaluation:**
   - I evaluated the model's performance on the test data using accuracy metrics.
   - The accuracy score, classification report, and confusion matrix were generated to assess the model's predictive capabilities.

In [30]:
```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Drop rows with missing values in relevant columns
merged_df_cleaned = merged_df.dropna(subset=['Wellness - Used_x', 'Age_x', 'Vacation - Used_x', 'Employee Status_x'])

# Define predictors (X) and target (y) after handling missing values
X = merged_df_cleaned[['Wellness - Used_x', 'Age_x', 'Vacation - Used_x']]
y = (merged_df_cleaned['Employee Status_x'] == 'Active').astype(int)  # Convert 'Active' to 1, 'Terminated' to 0

# Split 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)

# Initialize Random Forest Classifier
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)

# Fit the model on training data
rf_classifier.fit(X_train, y_train)

# Make predictions on test data
y_pred = rf_classifier.predict(X_test)

# Evaluate model performance
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# Generate classification report and confusion matrix
print("\nClassification Report:")
print(classification_report(y_test, y_pred))

print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

```
Accuracy: 0.9214226633581473

Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.90      0.89       426
           1       0.94      0.93      0.94       783

    accuracy                           0.92      1209
   macro avg       0.91      0.92      0.91      1209
weighted avg       0.92      0.92      0.92      1209


Confusion Matrix:
[[383  43]
 [ 52 731]]
```

Results

- **Accuracy:** 0.92 (rounded)
- **Confusion Matrix:** ((383 43) (52 731))

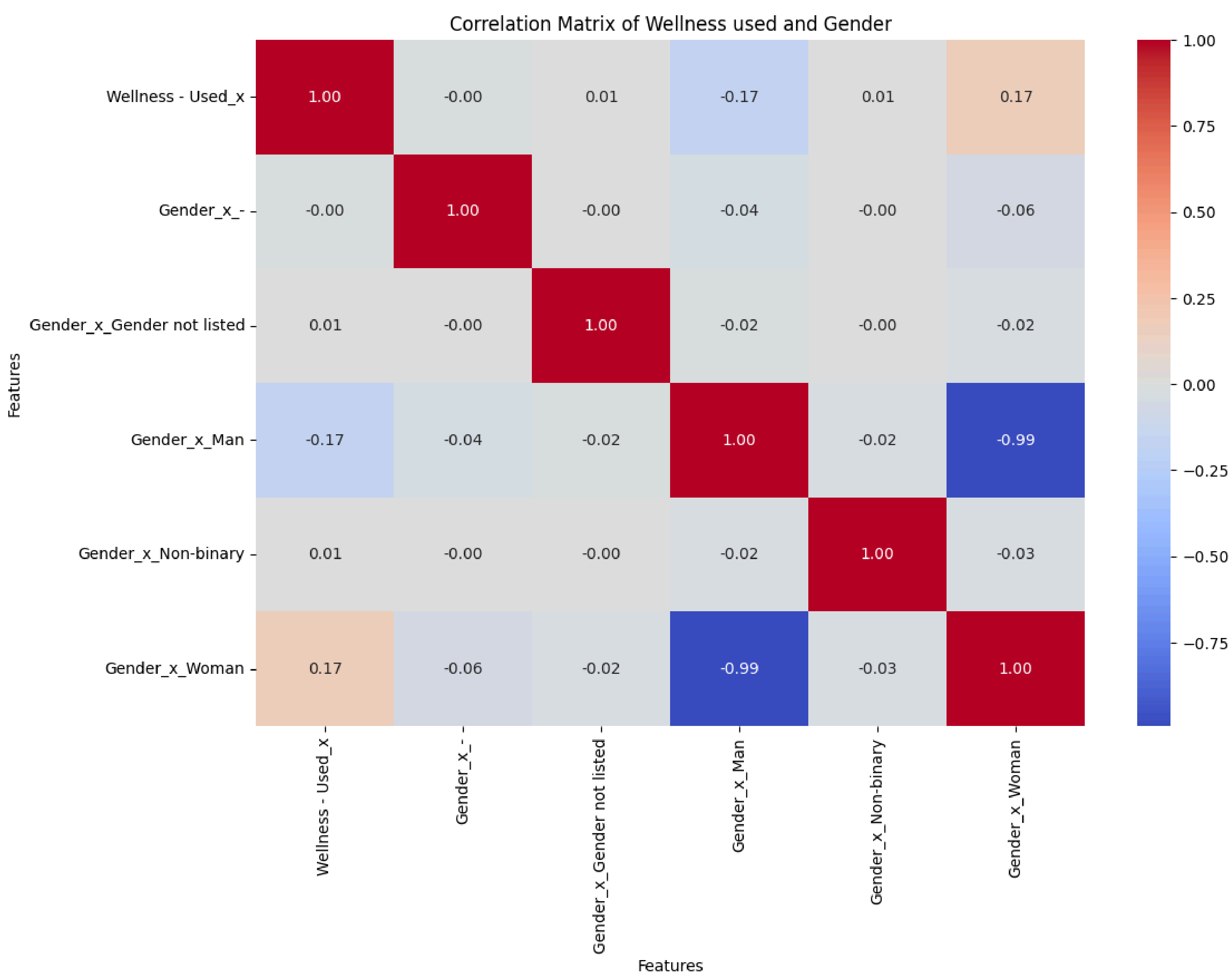
Conclusion

The Random Forest Classifier model provides insights into the factors influencing employee retention. Further analysis and feature engineering can enhance the model's predictive power and provide actionable insights for employee management strategies.

In [31]:
```python
import pandas as pd
df_1 = pd.read_excel('/kaggle/input/wellness2/2.xlsx', sheet_name = 'Diversity Data')

df_2 = pd.read_excel('/kaggle/input/wellness2/2.xlsx', sheet_name = '2023 Overtime Hours')

df_3 = pd.read_excel('/kaggle/input/wellness2/2.xlsx', sheet_name = '2022 Overtime Hours')

df_4 = pd.read_excel('/kaggle/input/wellness2/2.xlsx', sheet_name = '2022 Attrition Data')

df_5 = pd.read_excel('/kaggle/input/wellness2/2.xlsx', sheet_name = '2023 Attrition Data')

df_6 = pd.read_excel('/kaggle/input/wellness2/2.xlsx', sheet_name = '2023 Talent Review Data')

df_7 = pd.read_excel('/kaggle/input/wellness2/2.xlsx', sheet_name = '2023 Performance Data')

df_8 = pd.read_excel('/kaggle/input/wellness2/2.xlsx', sheet_name = '2022 Performance Data')
```

CORRELATION METRIX

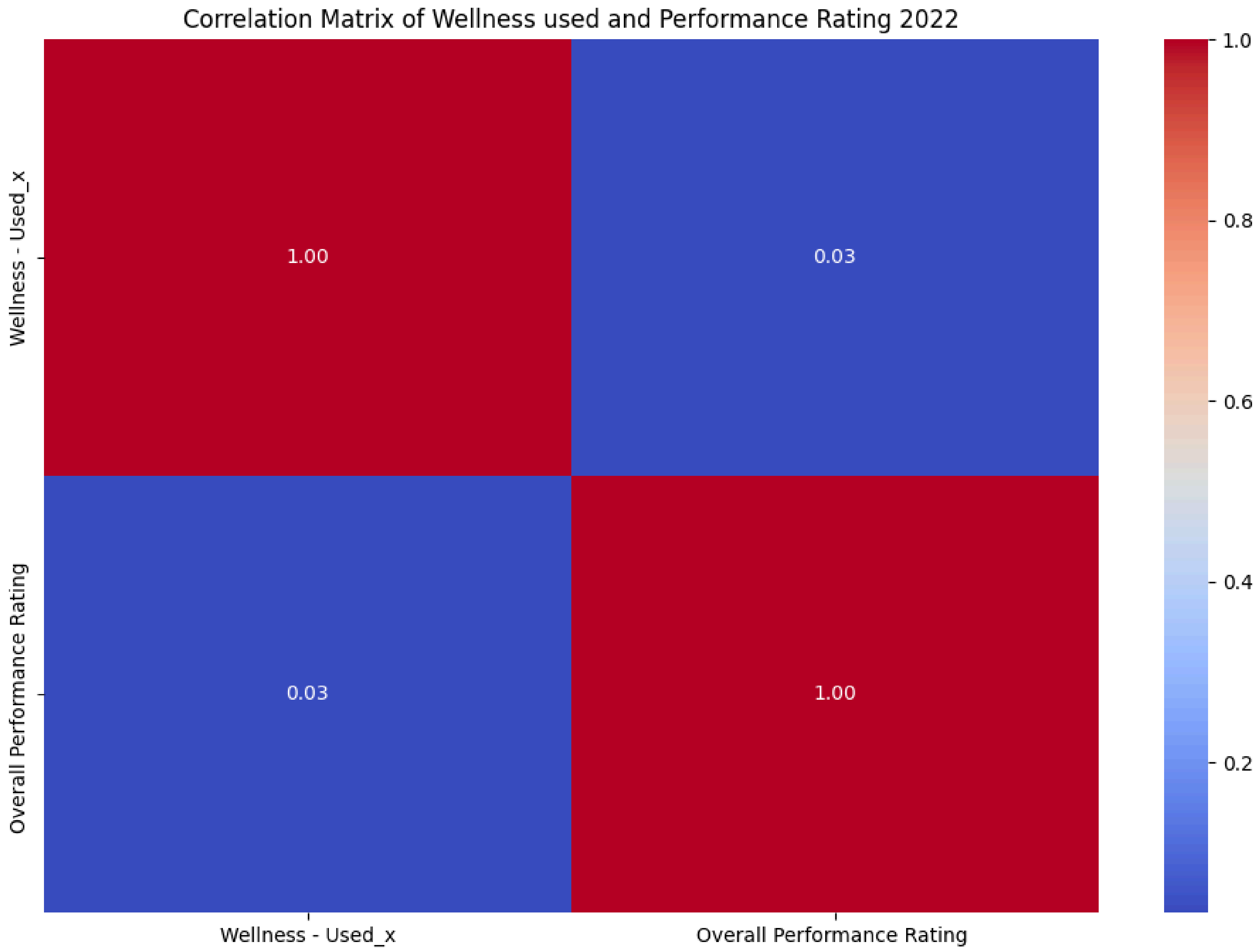
In [32]:
```python
# Convert categorical data to numerical using one-hot encoding
df_encoded = pd.get_dummies(merged_df_cleaned[['Wellness - Used_x', 'Gender_x']])

corr_matrix_encoded = df_encoded.corr()

# Plot the correlation matrix as a heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(corr_matrix_encoded, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix of Wellness used and Gender')
plt.xlabel('Features')
plt.ylabel('Features')
plt.show()
```



In [33]:
```python
import seaborn as sns
import matplotlib.pyplot as plt



df_combined = pd.DataFrame()
df_combined['Wellness - Used_x'] = merged_df['Wellness - Used_x']
df_combined['Overall Performance Rating '] = df_8['Overall Performance Rating']


# Replace 'unrate' with NaN values in the DataFrame
df_combined.replace('Unrated', pd.NA, inplace=True)

# Convert columns to numeric (if needed) to handle NaN values
df_combined = df_combined.apply(pd.to_numeric, errors='ignore')
```

```
# Drop rows with NaN values in any column
df_combined.dropna(inplace=True)

# Compute the correlation matrix for the combined DataFrame
corr_matrix_combined = df_combined.corr()


# Plot the correlation matrix as a heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(corr_matrix_combined, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix of Wellness used and Performance Rating 2022')
plt.show()
```

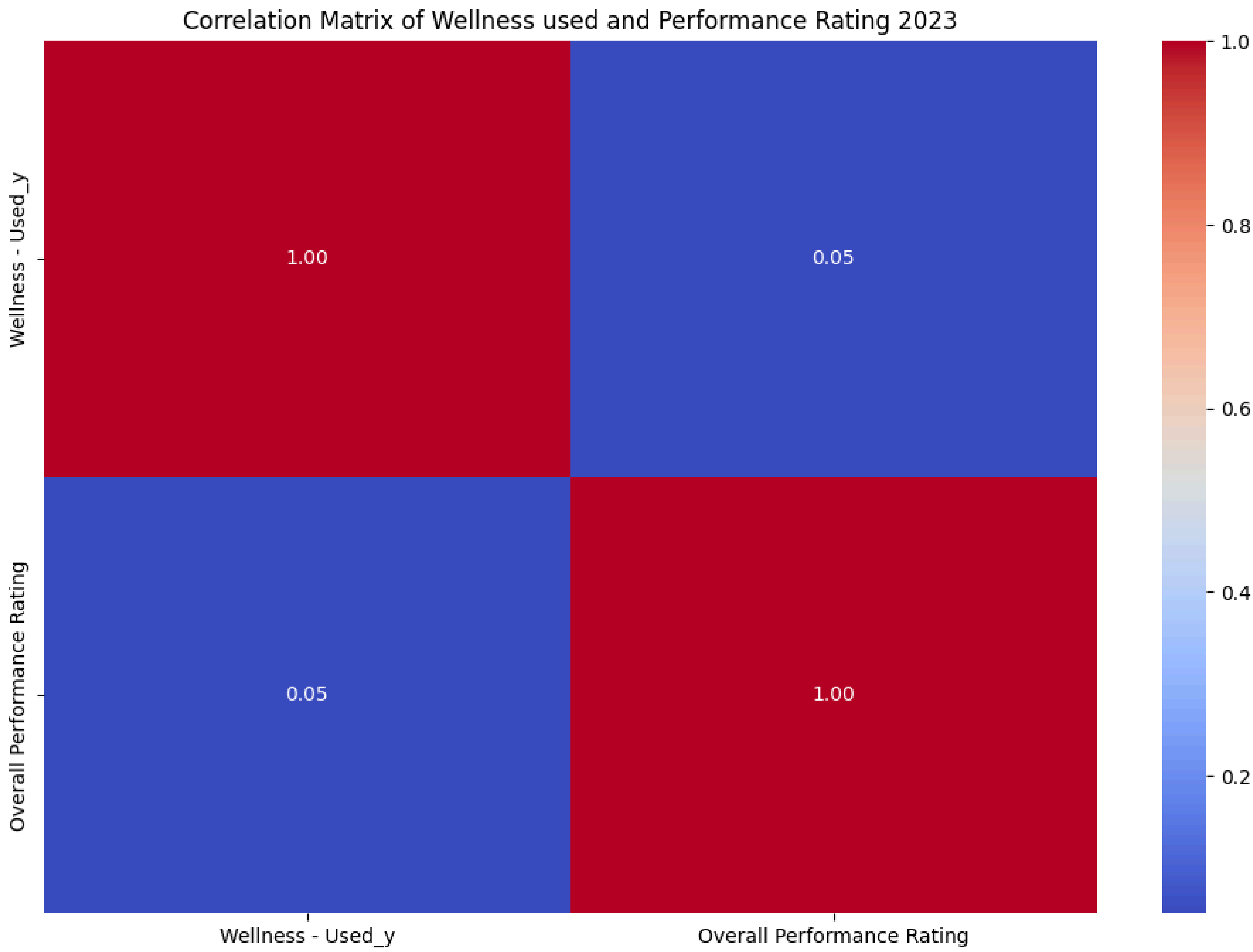
```
/tmp/ipykernel_18/467953103.py:15: FutureWarning: errors='ignore' is deprecated and will raise in a future version. Use to_numeric without passing `errors` and catch exceptions explicitly instead
  df_combined = df_combined.apply(pd.to_numeric, errors='ignore')
```



In [34]:

```
import seaborn as sns
import matplotlib.pyplot as plt



df_combined = pd.DataFrame()
df_combined['Wellness - Used_y'] = merged_df['Wellness - Used_y']
df_combined['Overall Performance Rating'] = df_7['Overall Performance Rating']


# Replace 'unrate' with NaN values in the DataFrame
df_combined.replace('Unrated', pd.NA, inplace=True)

# Convert columns to numeric (if needed) to handle NaN values
df_combined = df_combined.apply(pd.to_numeric, errors='ignore')

# Drop rows with NaN values in any column
df_combined.dropna(inplace=True)

# Compute the correlation matrix for the combined DataFrame
corr_matrix_combined = df_combined.corr()


# Plot the correlation matrix as a heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(corr_matrix_combined, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix of Wellness used and Performance Rating 2023')
plt.show()
```

```
/tmp/ipykernel_18/1396843458.py:15: FutureWarning: errors='ignore' is deprecated and will raise in a future version. Use to_numeric without passing `errors` and catch exceptions explicitly instead
  df_combined = df_combined.apply(pd.to_numeric, errors='ignore')
```


Correlation Matrix of Wellness used and Performance Rating 2023
Wellness - Used_y
Overall Performance Rating
1.00
0.05
0.05
1.00
Wellness - Used_y
Overall Performance Rating
1.0
0.8
0.6
0.4
0.2