In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
pd.set_option('display.max_columns', 50)
```

In [2]:
```
from IPython.display import Image, display

display(Image(filename=r'C:\Users\user\Downloads\bird-strike-story.jpg'))
```

A bird strike is strictly defined as a collision between a bird and an aircraft which is in flight or on a take-off or landing roll. The term is often expanded to cover other wildlife strikes - with bats or ground animals. Bird Strike is common and can be a significant threat to aircraft safety.Transport and communication are in the crucial domain in the field of analytics.Environmental impacts and safety are, nowadays, two major concerns of the scientific community with respect to transport scenarios and to the ever-growing urban areas. These issues gain more importance due to the increasing amount of vehicles and people. Seeking new solutions is reaching a point where available technologies andartificial intelligence, especially MAS, are being recognized as ways to cope with and tackle these kinds of problems in a distributed and more appropriate way. For smaller aircraft, significant damage may be caused to the aircraft structure and all aircraft, especially jet-engine

ones, are vulnerable to the loss of thrust which can follow the ingestion of birds into engine air intakes. This has resulted in several fatal accidents.

Bird strikes may occur during any phase of flight, but are most likely during the take-off, initial climb, approach and landing phases due to the greater numbers of birds in flight at lower levels. To have a closer look the following document visually depicts the data collected on Bird Strikes by FAA between 2000-2011

In [3]:
```
df=pd.read_csv("Bird Strikes data.xlsx - Bird Strikes.csv")
```

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

Out[4]:

	Record ID	Aircraft: Type	Airport: Name	Altitude bin	Aircraft: Make/Model	Wildlife: Number struck	Wildlife: Number Struck Actual	Effect: Impact to flight	FlightDate	Effect: Indicated Damage	Aircraft: Number of engines?	Aircr Airline/Opera
0	202152	Airplane	LAGUARDIA NY	> 1000 ft	B-737-400	Over 100	859	Engine Shut Down	11/23/00 0:00	Caused damage	2	US AIRWA
1	208159	Airplane	DALLAS/FORT WORTH INTL ARPT	< 1000 ft	MD-80	Over 100	424	NaN	7/25/01 0:00	Caused damage	2	AMERIC AIRLI
2	207601	Airplane	LAKEFRONT AIRPORT	< 1000 ft	C-500	Over 100	261	NaN	9/14/01 0:00	No damage	2	BUSIN
3	215953	Airplane	SEATTLE-TACOMA INTL	< 1000 ft	B-737-400	Over 100	806	Precautionary Landing	9/5/02 0:00	No damage	2	ALA AIRLI
4	219878	Airplane	NORFOLK INTL	< 1000 ft	CL-RJ100/200	Over 100	942	NaN	6/23/03 0:00	No damage	2	COM AIRLI

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

Out[5]:
```
(25558, 26)
```

In [6]:
```
df.columns
```

Out[6]:
```
Index(['Record ID', 'Aircraft: Type', 'Airport: Name', 'Altitude bin',
       'Aircraft: Make/Model', 'Wildlife: Number struck',
       'Wildlife: Number Struck Actual', 'Effect: Impact to flight',
       'FlightDate', 'Effect: Indicated Damage',
       'Aircraft: Number of engines?', 'Aircraft: Airline/Operator',
       'Origin State', 'When: Phase of flight', 'Conditions: Precipitation',
       'Remains of wildlife collected?',
       'Remains of wildlife sent to Smithsonian', 'Remarks', 'Wildlife: Size',
       'Conditions: Sky', 'Wildlife: Species',
       'Pilot warned of birds or wildlife?', 'Cost: Total $',
       'Feet above ground', 'Number of people injured', 'Is Aircraft Large?'],
      dtype='object')
```

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25558 entries, 0 to 25557
Data columns (total 26 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   Record ID                               25558 non-null  int64
 1   Aircraft: Type                          25429 non-null  object
 2   Airport: Name                           25429 non-null  object
 3   Altitude bin                            25429 non-null  object
 4   Aircraft: Make/Model                    25558 non-null  object
 5   Wildlife: Number struck                 25429 non-null  object
 6   Wildlife: Number Struck Actual          25558 non-null  int64
 7   Effect: Impact to flight                2078 non-null   object
 8   FlightDate                              25429 non-null  object
 9   Effect: Indicated Damage                25558 non-null  object
 10  Aircraft: Number of engines?            25291 non-null  object
 11  Aircraft: Airline/Operator              25429 non-null  object
 12  Origin State                            25109 non-null  object
 13  When: Phase of flight                   25429 non-null  object
 14  Conditions: Precipitation               2015 non-null   object
 15  Remains of wildlife collected?          25558 non-null  bool
 16  Remains of wildlife sent to Smithsonian 25558 non-null  bool
 17  Remarks                                 20787 non-null  object
 18  Wildlife: Size                          25429 non-null  object
 19  Conditions: Sky                         25558 non-null  object
 20  Wildlife: Species                       25558 non-null  object
 21  Pilot warned of birds or wildlife?      25429 non-null  object
 22  Cost: Total $                           25558 non-null  object
 23  Feet above ground                       25429 non-null  object
 24  Number of people injured                25558 non-null  int64
 25  Is Aircraft Large?                      25429 non-null  object
dtypes: bool(2), int64(3), object(21)
memory usage: 4.7+ MB
```

In [8]:
```
df.dtypes
```

Out[8]:
```
Record ID                                int64
Aircraft: Type                          object
Airport: Name                           object
Altitude bin                            object
Aircraft: Make/Model                    object
Wildlife: Number struck                 object
Wildlife: Number Struck Actual           int64
Effect: Impact to flight                object
FlightDate                              object
Effect: Indicated Damage                object
Aircraft: Number of engines?            object
Aircraft: Airline/Operator              object
Origin State                            object
When: Phase of flight                   object
Conditions: Precipitation               object
Remains of wildlife collected?            bool
Remains of wildlife sent to Smithsonian   bool
Remarks                                 object
Wildlife: Size                          object
Conditions: Sky                         object
Wildlife: Species                       object
Pilot warned of birds or wildlife?      object
Cost: Total $                           object
Feet above ground                       object
Number of people injured                 int64
Is Aircraft Large?                      object
dtype: object
```

In [9]:
```
df['Year'] = pd.to_datetime(df['FlightDate']).dt.year
#df['Year']=df['FlightDate'].dt.year
df['Month']=pd.to_datetime(df['FlightDate']).dt.month
```

```
C:\Users\user\AppData\Local\Temp\ipykernel_9828\1400486001.py:1: UserWarning: Could not infer format, so each element will be p
arsed individually, falling back to `dateutil`. To ensure parsing is consistent and as-expected, please specify a format.
  df['Year'] = pd.to_datetime(df['FlightDate']).dt.year
C:\Users\user\AppData\Local\Temp\ipykernel_9828\1400486001.py:3: UserWarning: Could not infer format, so each element will be p
arsed individually, falling back to `dateutil`. To ensure parsing is consistent and as-expected, please specify a format.
  df['Month']=pd.to_datetime(df['FlightDate']).dt.month
```

In [10]:
```
df.describe()
```

Out[10]:

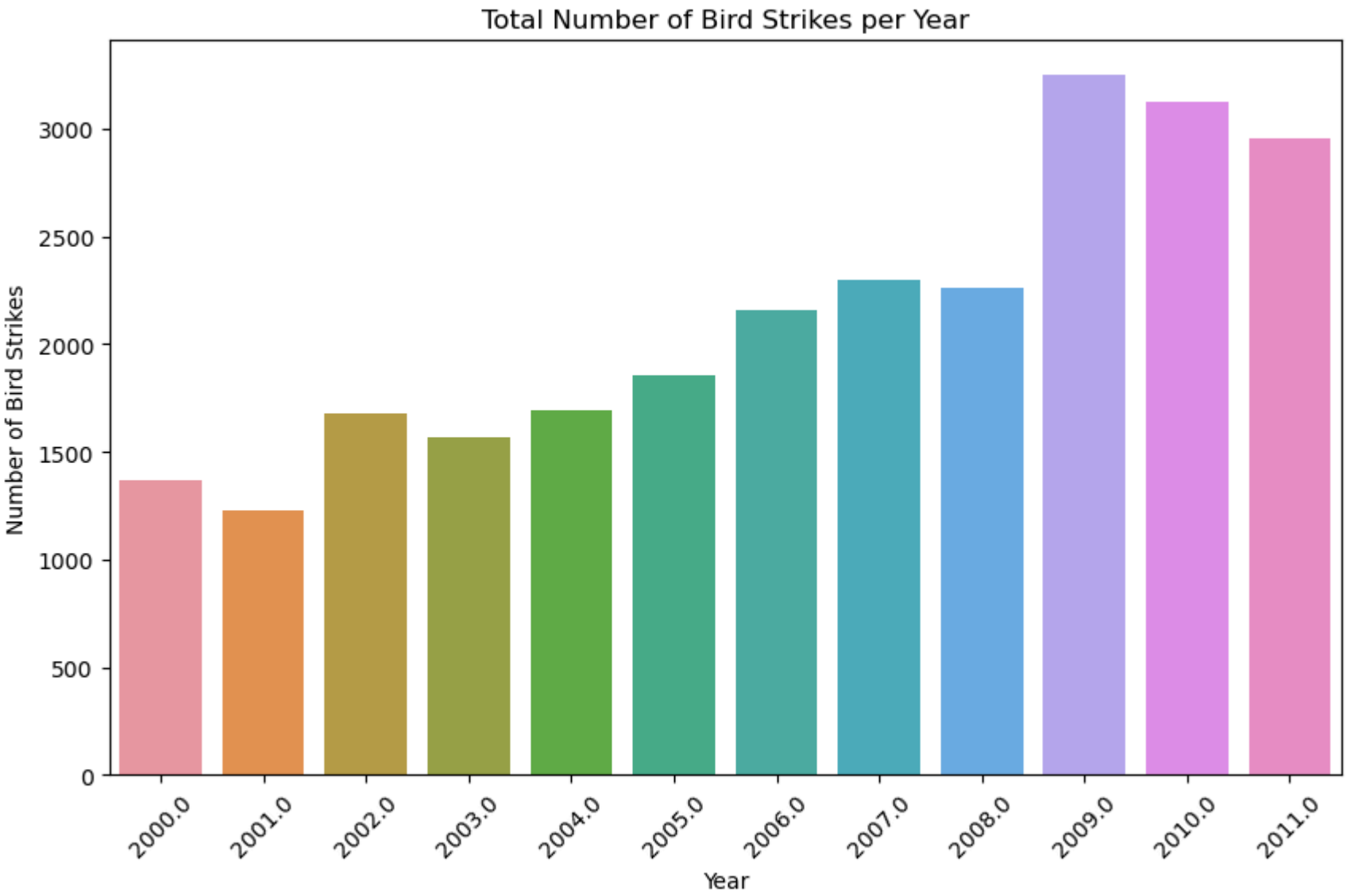
	Record ID	Wildlife: Number Struck Actual	Number of people injured	Year	Month
count	25558.000000	25558.000000	25558.000000	25429.000000	25429.000000
mean	253916.085609	2.691525	0.001056	2006.502772	7.210193
std	38510.453382	12.793975	0.050420	3.362241	2.793630
min	1195.000000	1.000000	0.000000	2000.000000	1.000000
25%	225783.750000	1.000000	0.000000	2004.000000	5.000000
50%	248749.000000	1.000000	0.000000	2007.000000	8.000000
75%	269168.750000	1.000000	0.000000	2009.000000	9.000000
max	321909.000000	942.000000	6.000000	2011.000000	12.000000

CASE STUDIES

- Visuals Depicting the Number of Bird Strikes

In [11]:

```
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='Year')
plt.title('Total Number of Bird Strikes per Year')
plt.xlabel('Year')
plt.ylabel('Number of Bird Strikes')
plt.xticks(rotation=45)
plt.show()
```

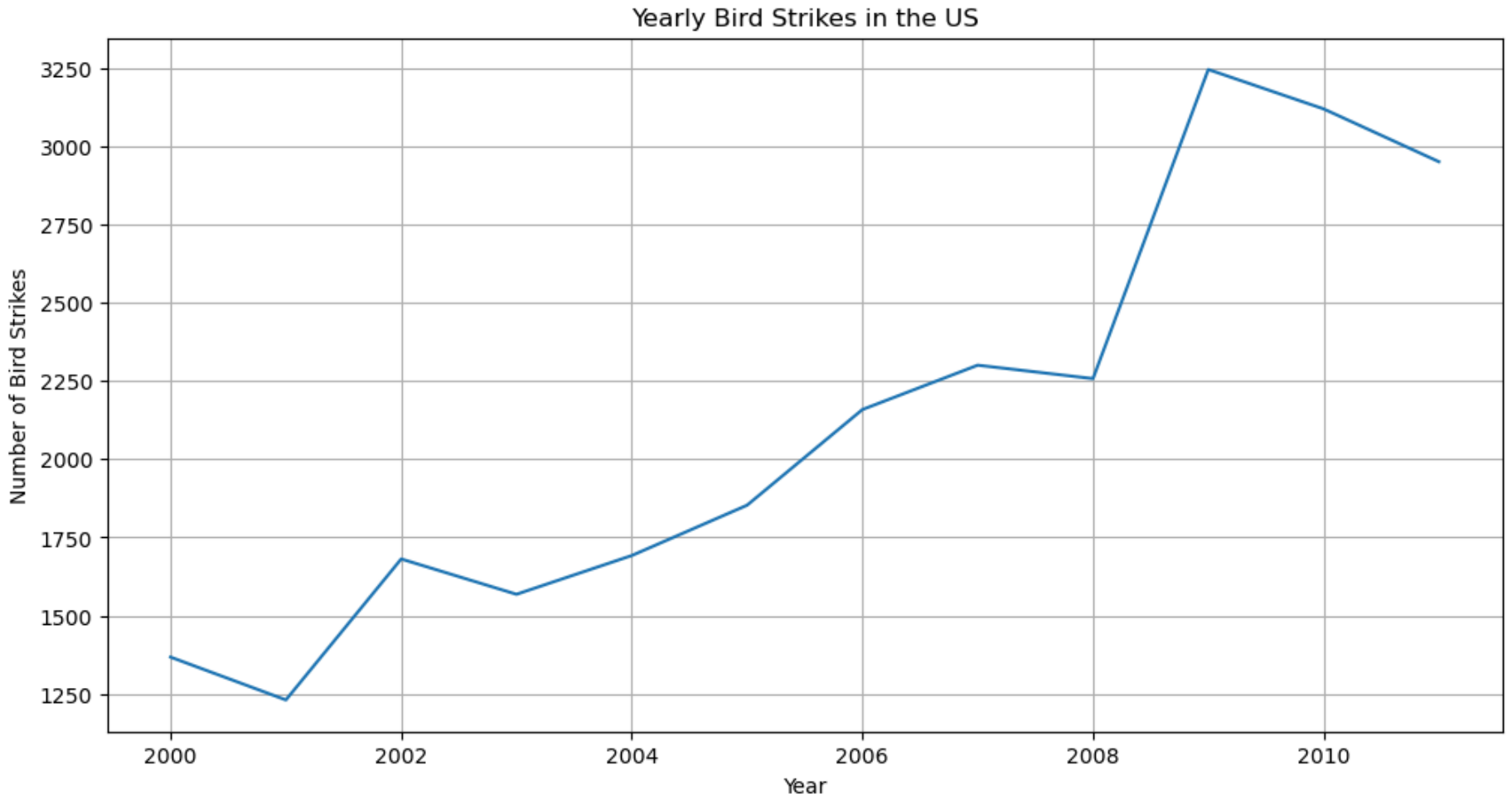


📌In the above bar graph displays the total number of bird strikes reported each year. Look for trends over time, such as increases or decreases in the number of strikes.

- Yearly Analysis & Bird Strikes in the US

In [12]:
```
yearly=df['Year'].value_counts().sort_index()
```

In [13]:
```
plt.figure(figsize=(12, 6))
yearly.plot(kind='line')
plt.title('Yearly Bird Strikes in the US')
plt.xlabel('Year')
plt.ylabel('Number of Bird Strikes')
plt.grid(True)
plt.show()
```

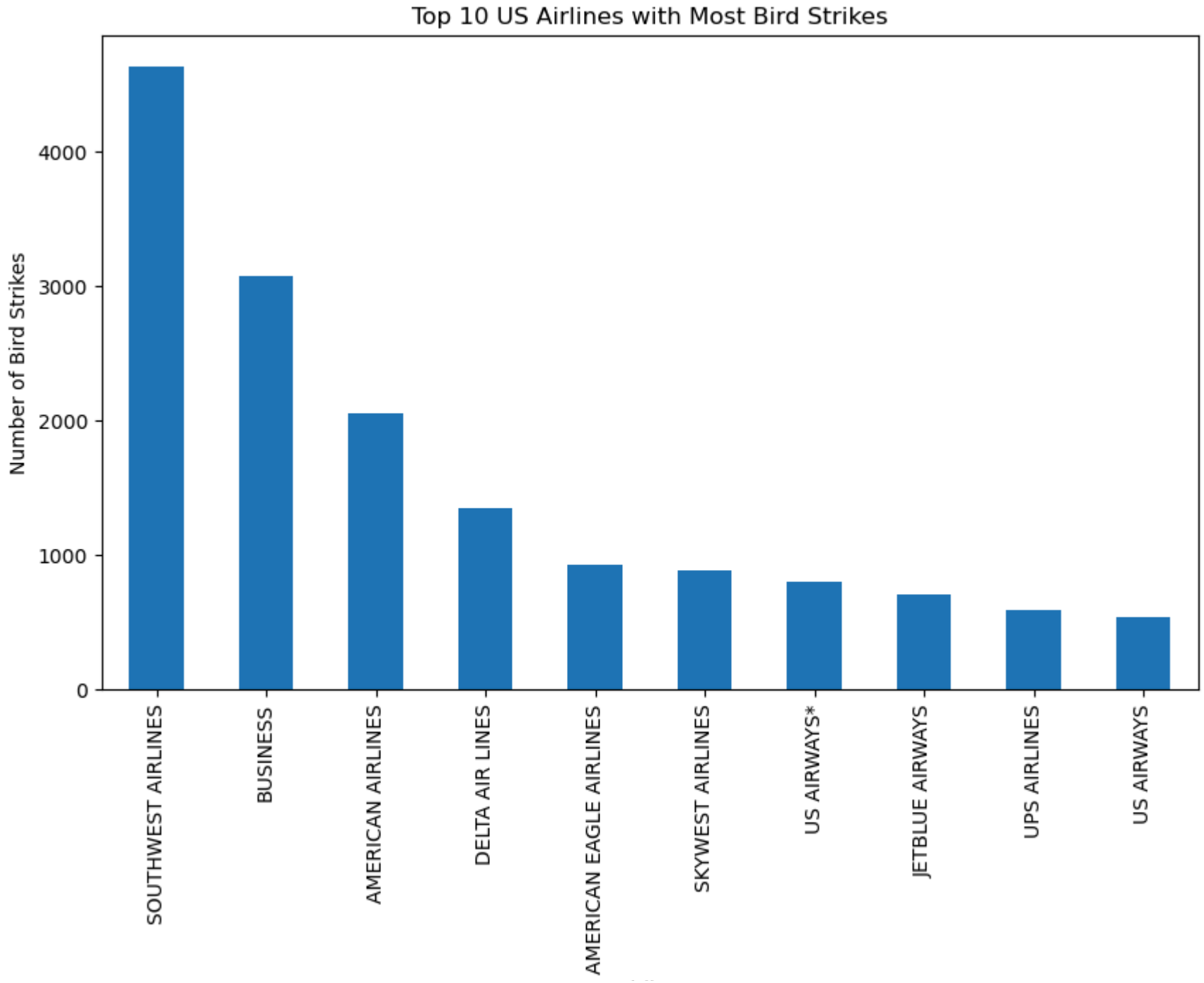


📌This above line graph shows the number of bird strikes in the US over the years. In year 2001 the bird strikes in US start increases and decreases , but in year 2009 its at pinpoint with notable strikes

● Top 10 US Airlines in Terms of Having Encountered Bird Strikes

In [14]:
```
top_airlines=df['Aircraft: Airline/Operator'].value_counts().head(10)
```

In [15]:
```
plt.figure(figsize=(10,6))
top_airlines.plot(kind='bar')
plt.title('Top 10 US Airlines with Most Bird Strikes')
plt.xlabel('Airline')
plt.ylabel('Number of Bird Strikes')
plt.xticks(rotation=90)
plt.show()
```


Top 10 US Airlines with Most Bird Strikes
Number of Bird Strikes
0
1000
2000
3000
4000
SOUTHWEST AIRLINES
BUSINESS
AMERICAN AIRLINES
DELTA AIR LINES
AMERICAN EAGLE AIRLINES
SKYWEST AIRLINES
US AIRWAYS*
JETBLUE AIRWAYS
UPS AIRLINES
US AIRWAYS

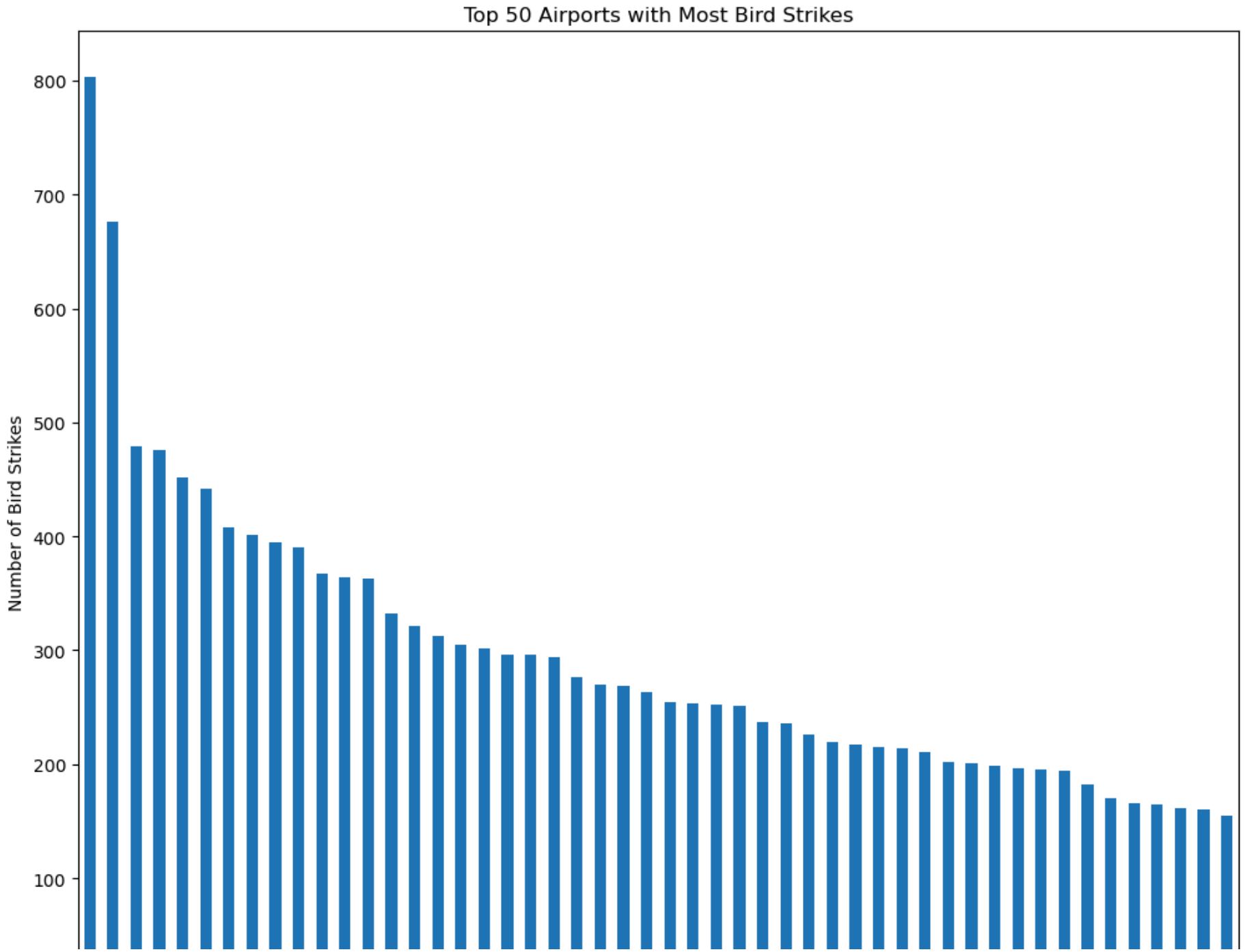
Airline

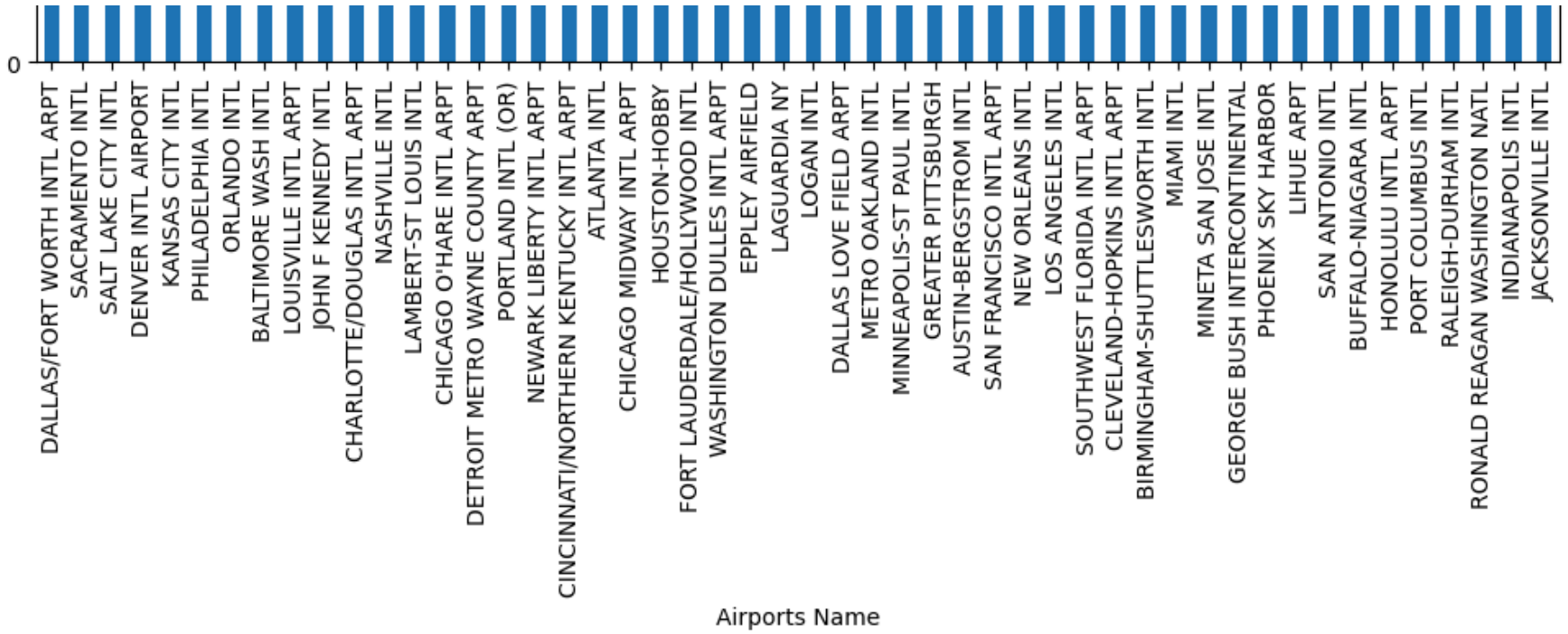
📌The above bar graph shows that Southwest Airlines had most affected by bird strikes

- Airports with most incidents of bird strikes – Top 50

In [16]:
```
top_airports=df['Airport: Name'].value_counts().head(50)
```

In [17]:
```
plt.figure(figsize=(12, 10))
top_airports.plot(kind='bar')
plt.title('Top 50 Airports with Most Bird Strikes')
plt.xlabel('Airports Name')
plt.ylabel('Number of Bird Strikes')
plt.xticks(rotation=90)
plt.show()
```


Top 50 Airports with Most Bird Strikes
Number of Bird Strikes
800
700
600
500
400
300
200
100



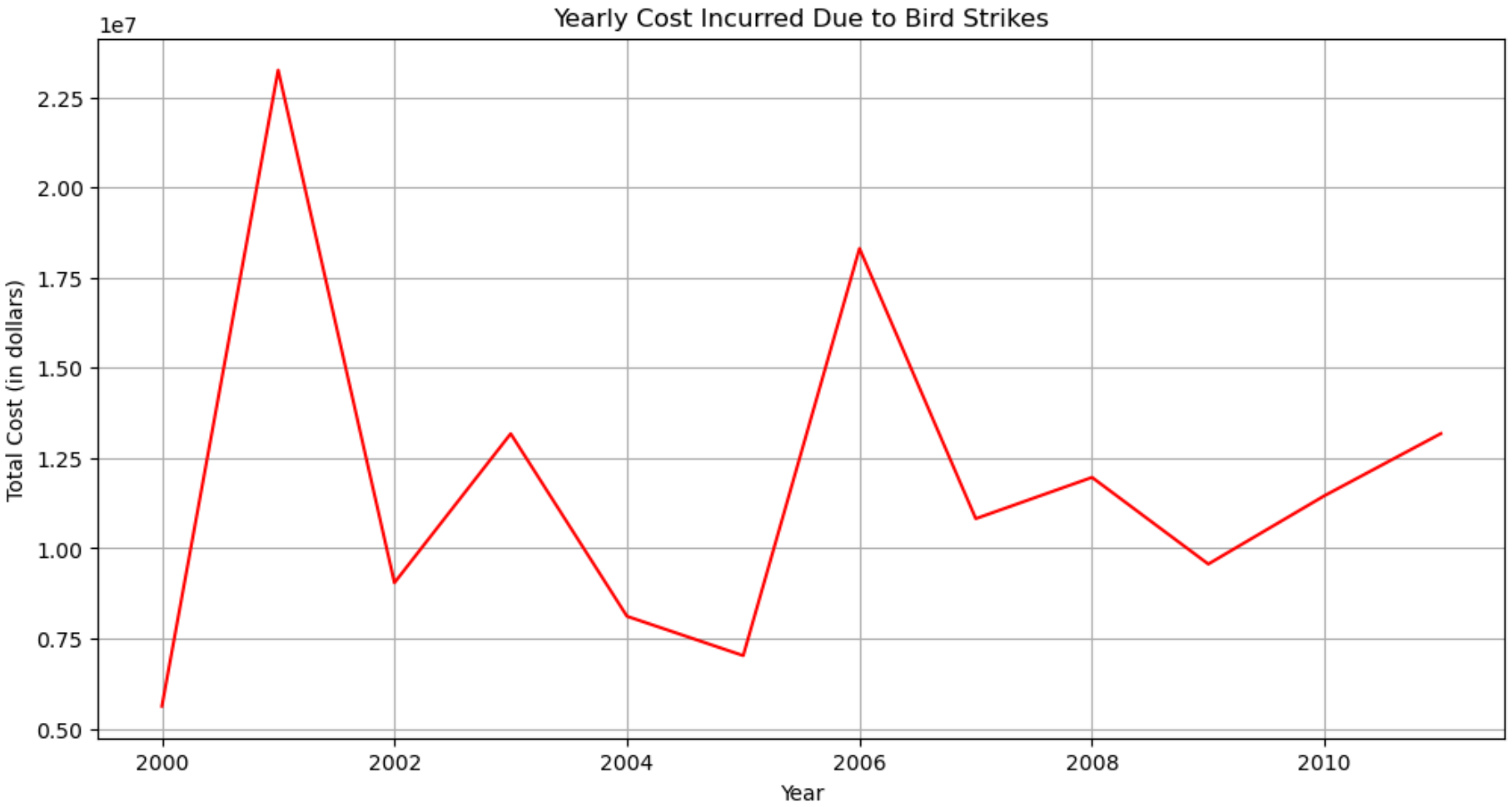
📌This bar graph represents the top 50 Airports in which DALLAS/FORT WORTH INTL ARPT with taller bars have experienced more bird strikes.

● Yearly Cost Incurred due to Bird Strikes:

In [18]:
```
df['Cost: Total $'] = df['Cost: Total $'].str.replace(',', '').astype(float, errors='ignore')
```

In [19]:
```
Yearly_Cost = df.groupby('Year')['Cost: Total $'].sum()

plt.figure(figsize=(12, 6))
Yearly_Cost.plot(kind='line',color='red')
plt.title('Yearly Cost Incurred Due to Bird Strikes')
plt.xlabel('Year')
plt.ylabel('Total Cost (in dollars)')
plt.grid(True)
plt.show()
print(f'The total cost incurred due to bird stike is:{Yearly_Cost}')
```


Yearly Cost Incurred Due to Bird Strikes
1e7
2.25
2.00
1.75
1.50
1.25
1.00
0.75
0.50
Total Cost (in dollars)
2000
2002
2004
2006
2008
2010
Year

```
The total cost incurred due to bird stike is:Year
2000.0     5625496.0
2001.0    23252168.0
2002.0     9046405.0
2003.0    13176787.0
2004.0     8116866.0
2005.0     7026670.0
2006.0    18309903.0
2007.0    10822426.0
2008.0    11966121.0
2009.0     9564327.0
2010.0    11459879.0
2011.0    13180130.0
Name: Cost: Total $, dtype: float64
```

📌The line shows how costs have changed over time. Look for years with high costs, indicating significant financial impacts due to bird strikes.
In 2001 most highest financial impacts due to bird strikes

- When do most bird strikes occur?

In [20]:
```
df['FlightDate']=pd.to_datetime(df['FlightDate'])
```

```
C:\Users\user\AppData\Local\Temp\ipykernel_9828\3281754392.py:1: UserWarning: Could not infer format, so each element will be p
arsed individually, falling back to `dateutil`. To ensure parsing is consistent and as-expected, please specify a format.
  df['FlightDate']=pd.to_datetime(df['FlightDate'])
```

In [21]:
```
df['hour'] = pd.to_datetime(df['FlightDate']).dt.hour
```

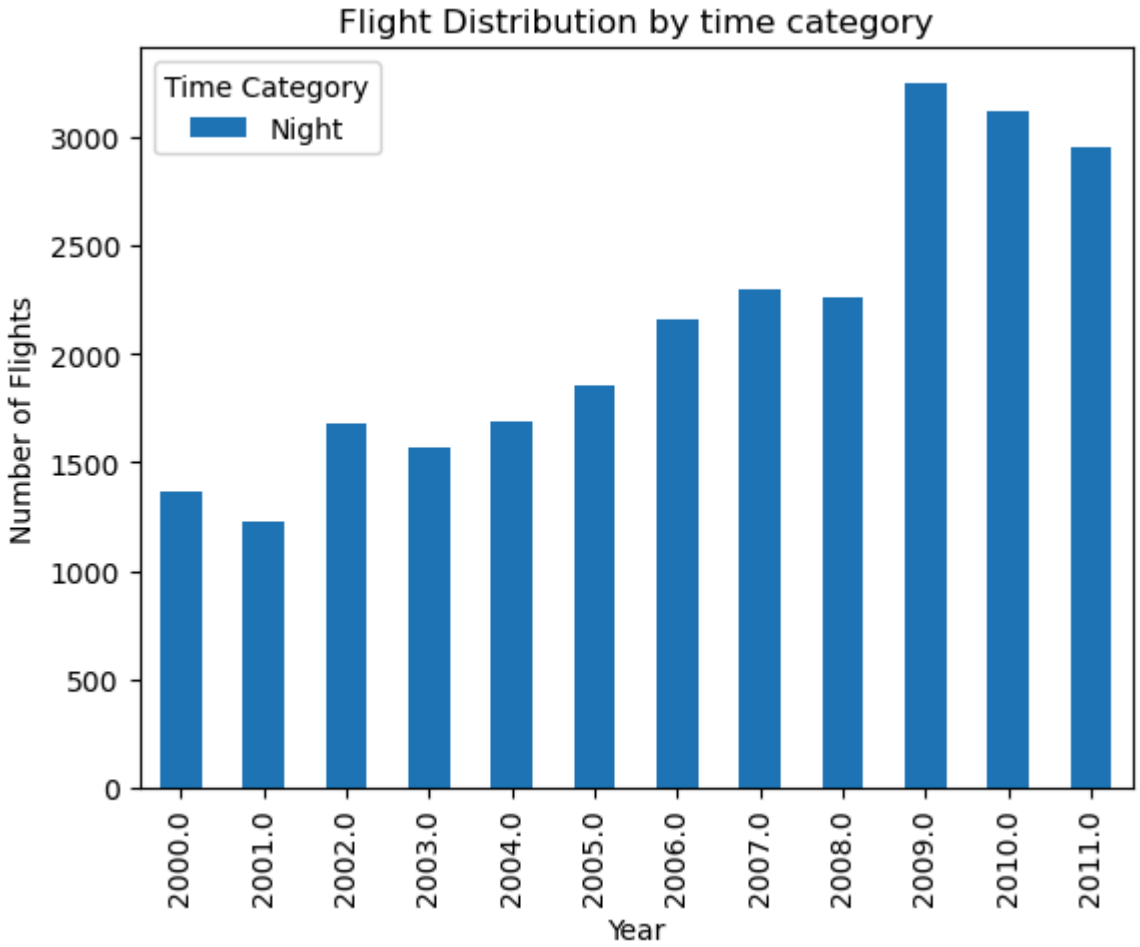
In [22]:
```
def categorize_time(hour):
    if 0<= hour <6:
        return "Night"
    elif 6<= hour<12:
        return "Morning"
    elif 12<= hour<18:
        return "Afternoon"
    else:
        return "Evening"
```

In [23]:
```
df["Time Category"]=df["hour"].apply(categorize_time)
```

In [24]:
```
#time_category_counts=df['Time Category'].value_counts()
```

In [25]:
```
time_category=df.groupby(['Year','Time Category']).size().unstack()
```

In [26]:
```
time_category.plot(kind='bar',stacked=True)
plt.xlabel("Year")
plt.ylabel("Number of Flights")
plt.title("Flight Distribution by time category")
plt.show()
```

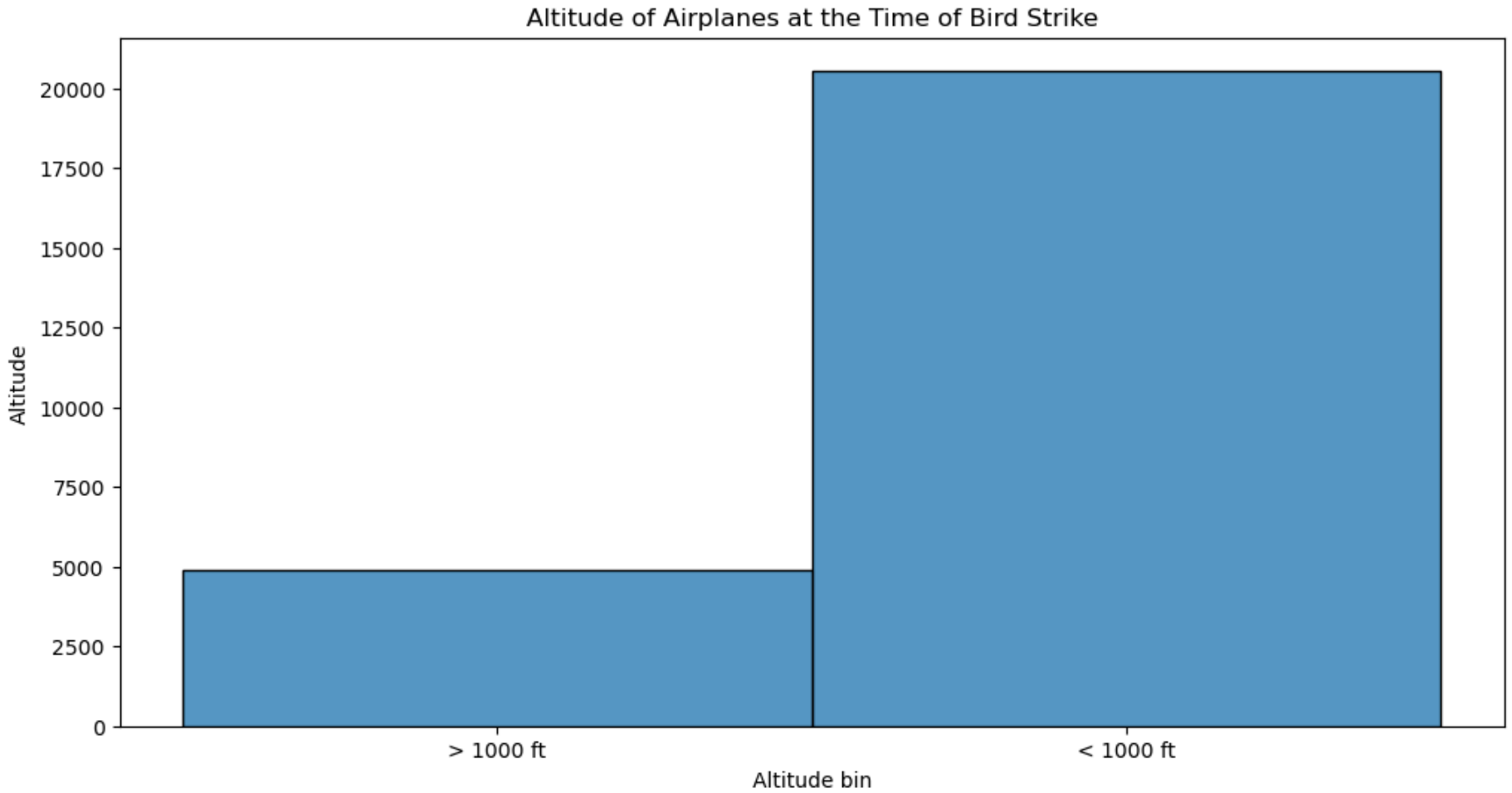


📌Most of bird strikes happened at night over all year

- Altitude of aeroplanes at the time of strike

In [27]:
```
plt.figure(figsize=(12, 6))
sns.histplot(data=df,x="Altitude bin",bins=50, kde=False,cbar=True)
plt.xlabel('Altitude bin')
plt.ylabel('Altitude')
plt.title('Altitude of Airplanes at the Time of Bird Strike')
plt.show()
```

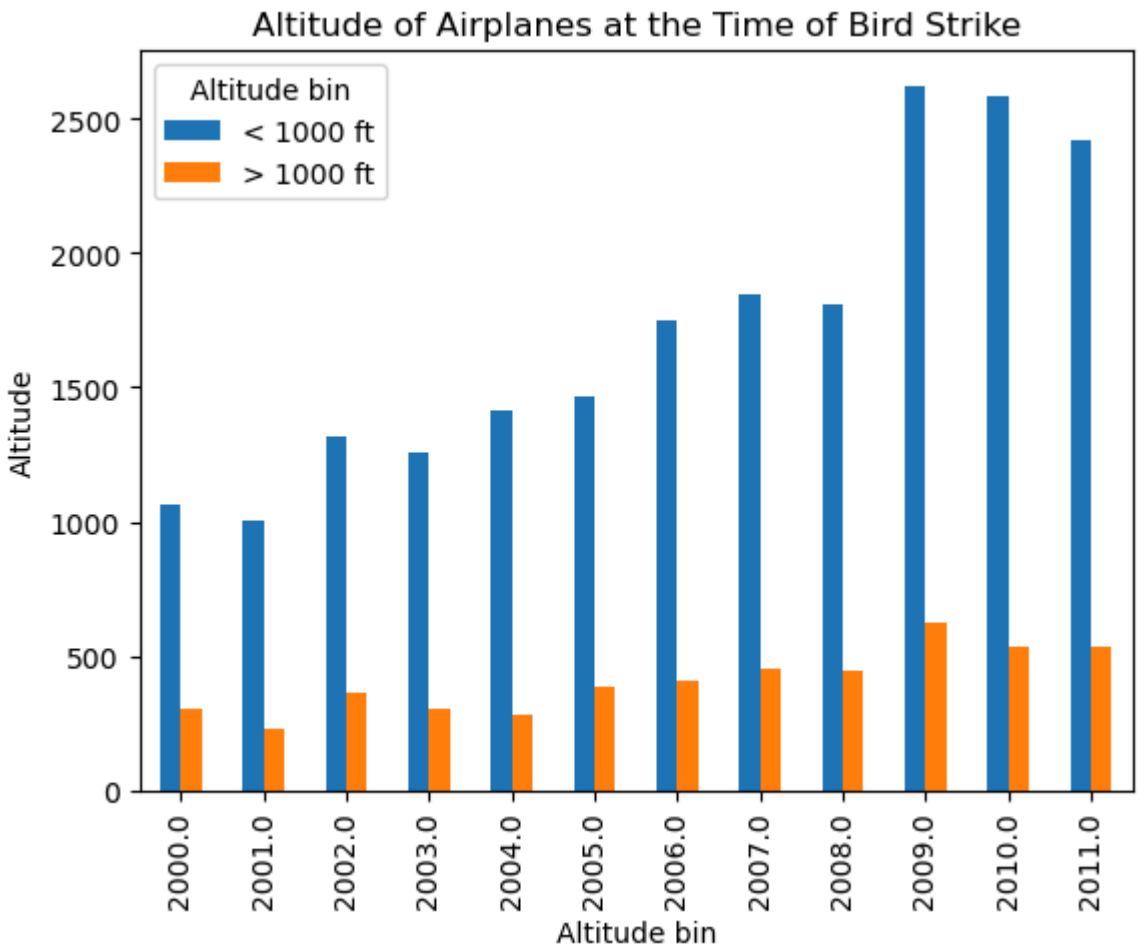
```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



In [28]:
```
year_altitude=df.groupby(['Year','Altitude bin']).size().unstack()
```

In [29]:
```
plt.figure(figsize=(12, 6))
year_altitude.plot(kind='bar')
plt.xlabel('Altitude bin')
plt.ylabel('Altitude')
plt.title('Altitude of Airplanes at the Time of Bird Strike')
plt.show()
```

```
<Figure size 1200x600 with 0 Axes>
```



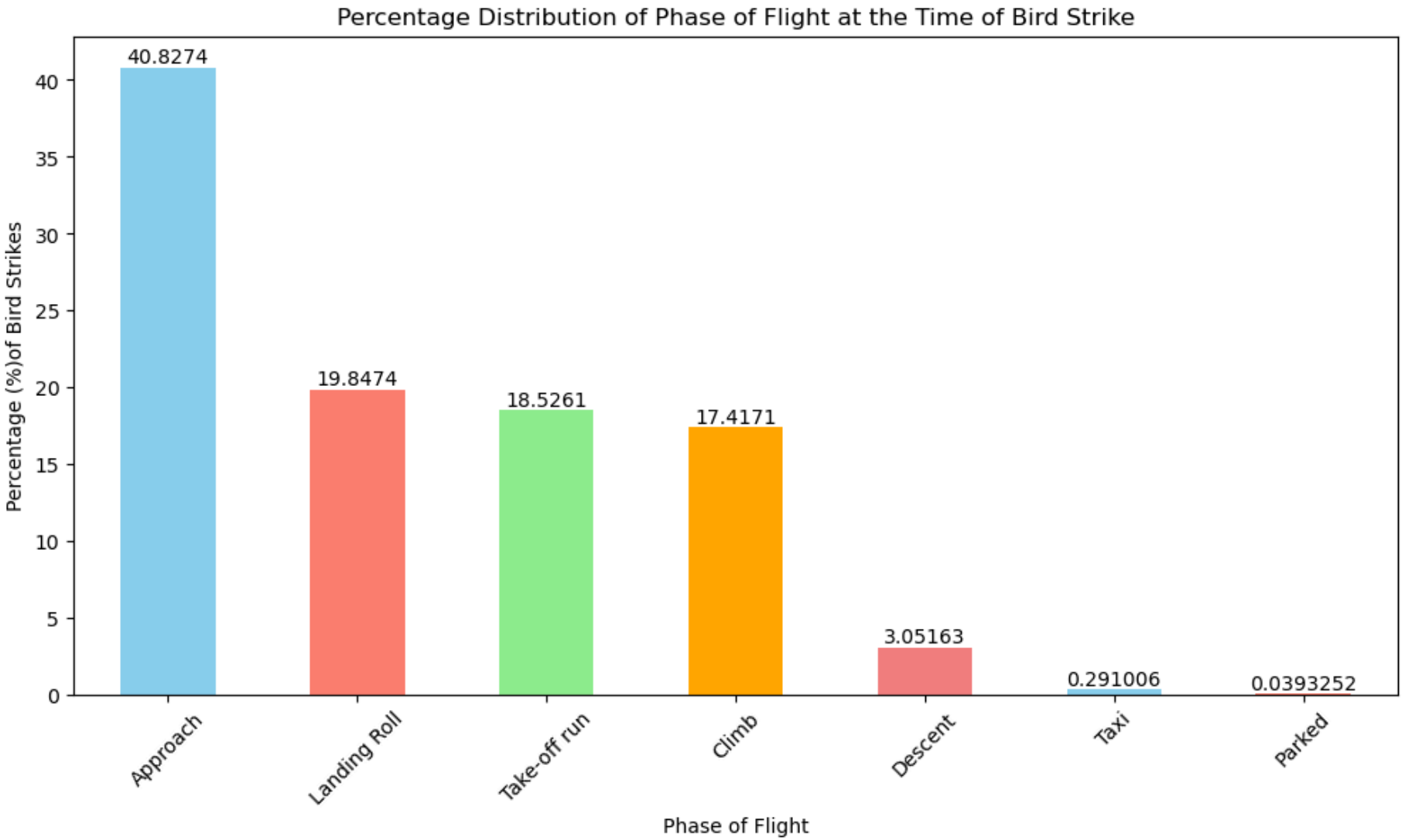
📌 20000 of bird strike incidents have happened when the altitude of airplane was <1000 ft and 5000 have happend when altitude was >1000 ft.

- Phase of flight at the time of the strike.

In [30]:
```
phase_of_flight_Percentage=df['When: Phase of flight'].value_counts(normalize=True)*100
```

In [31]:
```
colors=['skyblue','salmon','lightgreen','orange','lightcoral']
```

In [32]:
```
plt.figure(figsize=(12,6))
ax=phase_of_flight_Percentage.plot(kind='bar',color=colors)
ax.bar_label(ax.containers[0])
plt.xlabel('Phase of Flight')
plt.ylabel('Percentage (%)of Bird Strikes')
plt.title('Percentage Distribution of Phase of Flight at the Time of Bird Strike')
plt.xticks(rotation=45)
plt.show()
```

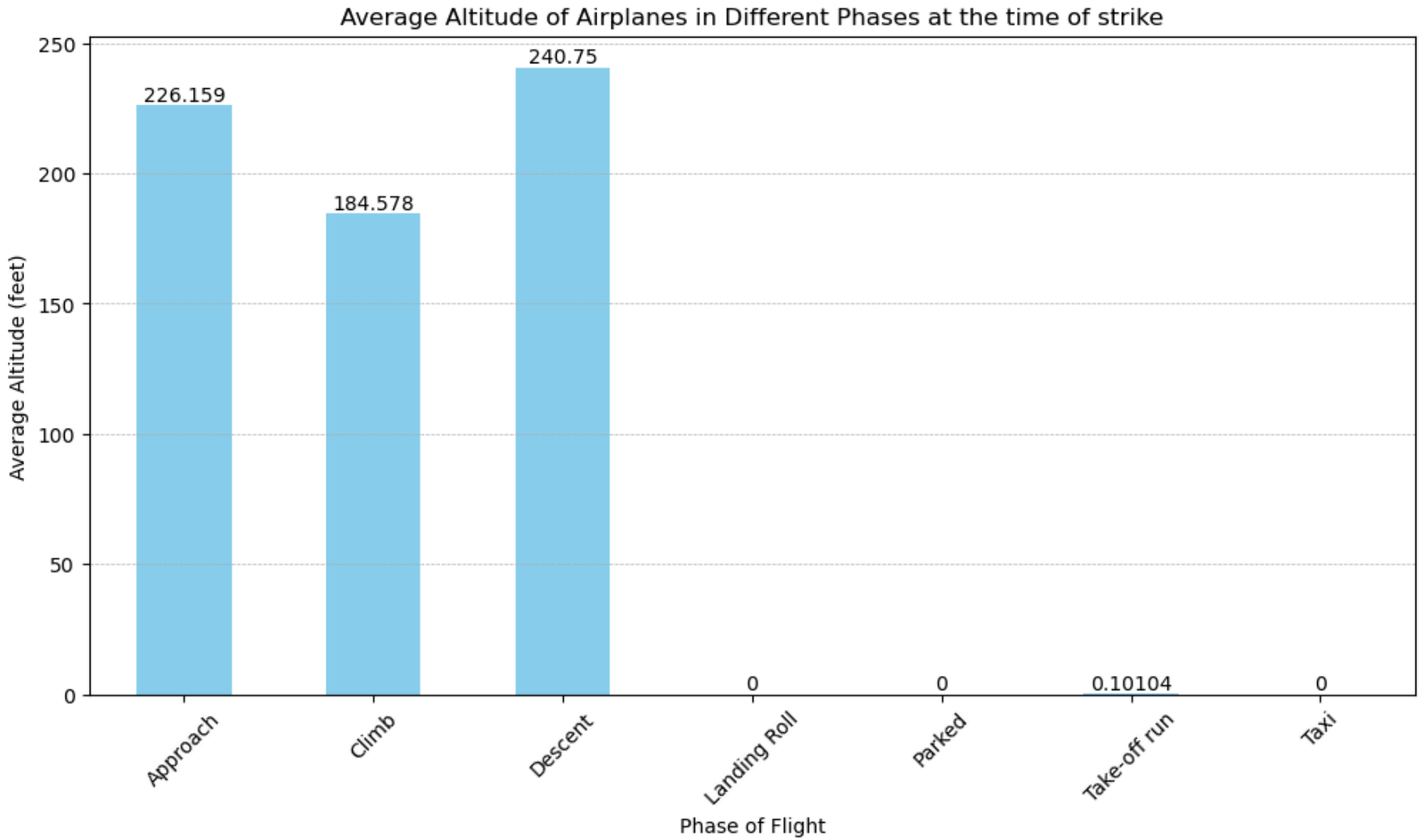


📌At the time of Approach Phase approx 40.83% prone to bird strikes.

- Average Altitude of the aeroplanes in different phases at the time of strike

In [33]:
```
df['Feet above ground']=pd.to_numeric(df['Feet above ground'],errors='coerce')
average_altitude = df.groupby('When: Phase of flight')['Feet above ground'].mean()
```

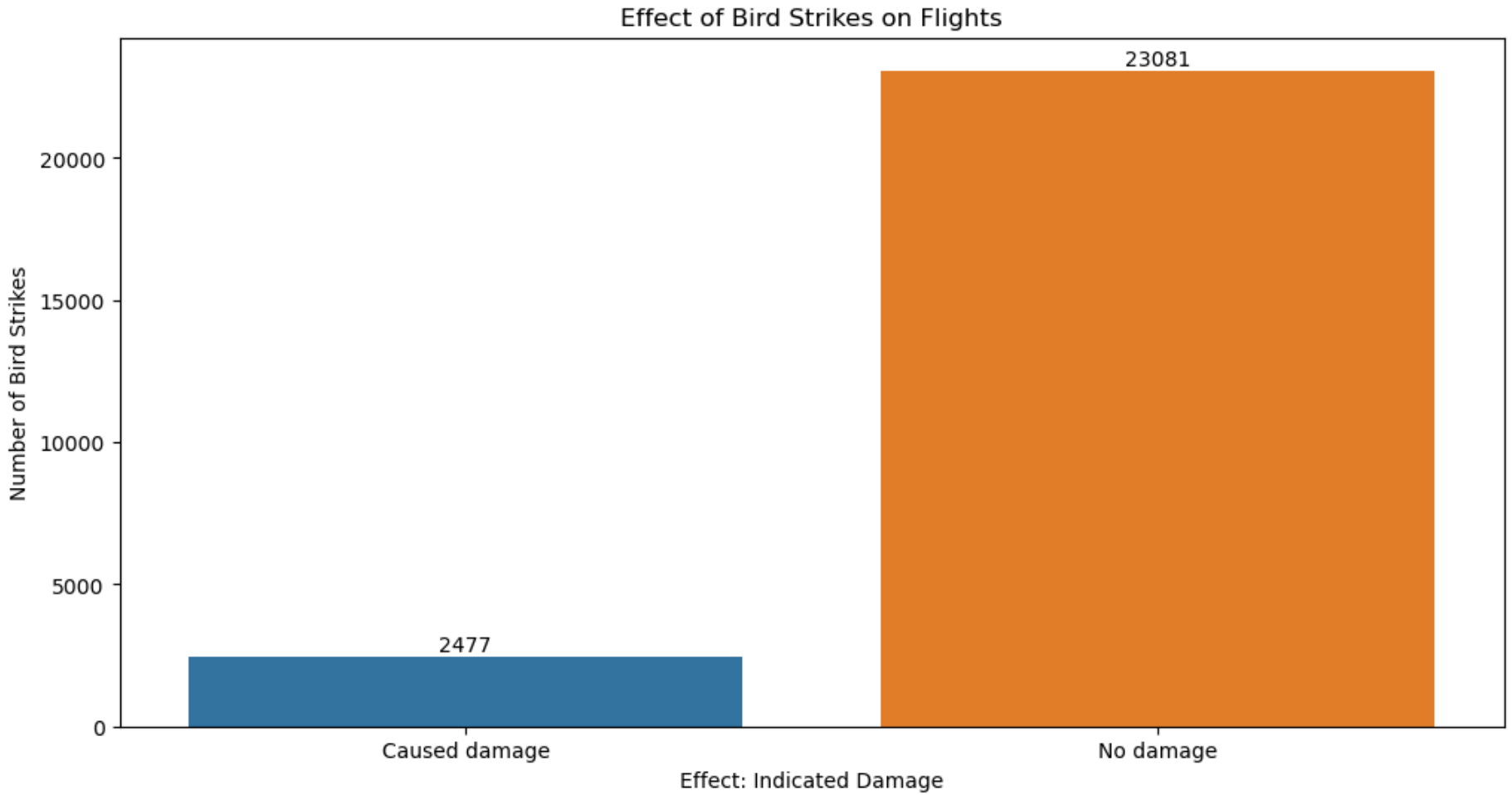
```
plt.figure(figsize=(12, 6))
ax=average_altitude.plot(kind='bar',color='skyblue')
ax.bar_label(ax.containers[0])
plt.title('Average Altitude of Airplanes in Different Phases at the time of strike')
plt.xlabel('Phase of Flight')
plt.ylabel('Average Altitude (feet)')
plt.xticks(rotation=45)
plt.grid(axis='y',linestyle='--',linewidth=0.5)
plt.show()
print("Average Altitude of Airplanes in Different Phases at the time of strike :",average_altitude )
```


Average Altitude of Airplanes in Different Phases at the time of strike
Average Altitude (feet)
250
200
150
100
50
0
226.159
184.578
240.75
0
0
0.10104
0
Approach
Climb
Descent
Landing Roll
Parked
Take-off run
Taxi
Phase of Flight

```
Average Altitude of Airplanes in Different Phases at the time of strike : When: Phase of flight
Approach        226.158614
Climb           184.578211
Descent         240.750000
Landing Roll      0.000000
Parked            0.000000
Take-off run      0.101040
Taxi              0.000000
Name: Feet above ground, dtype: float64
```

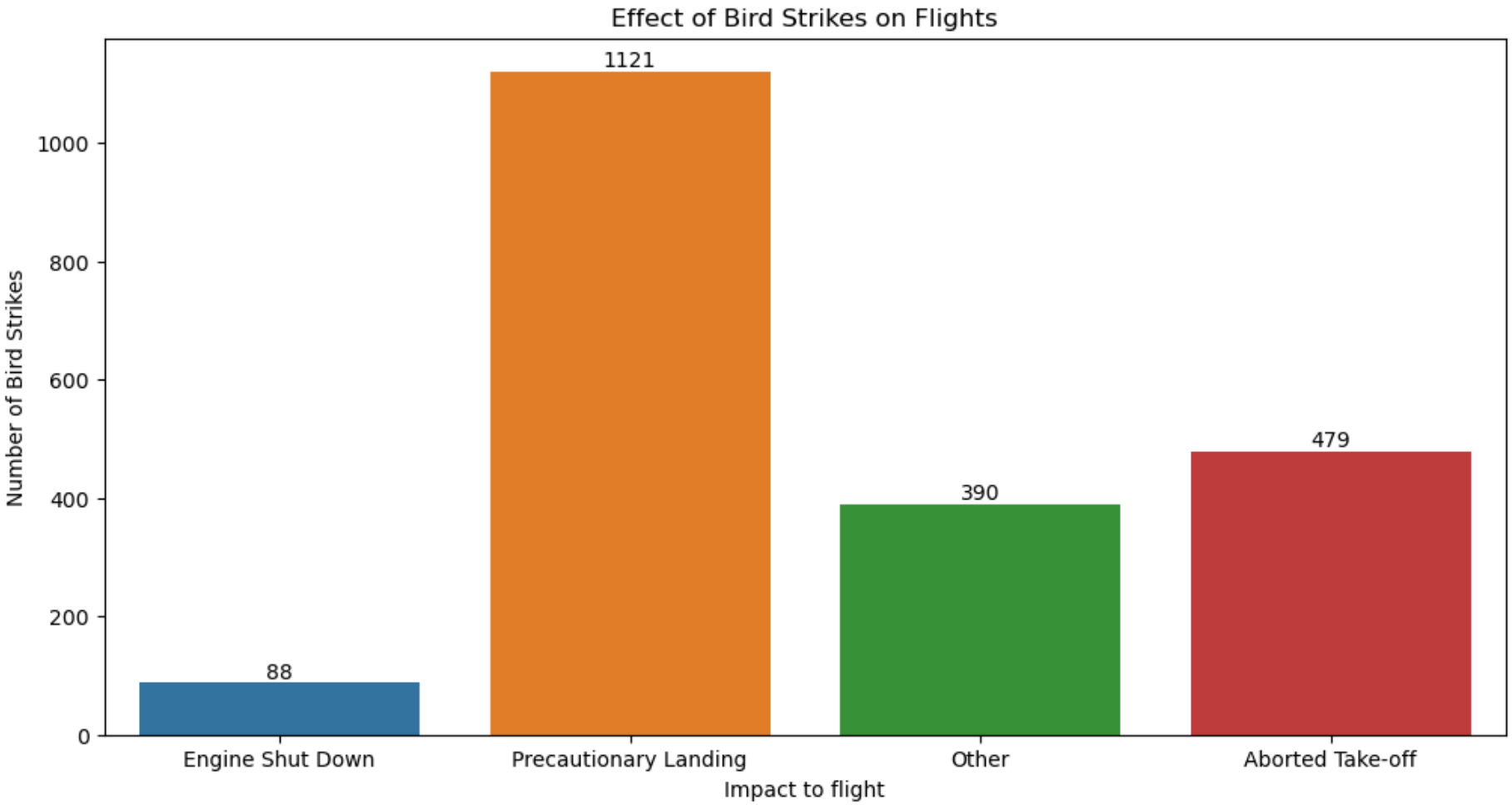
- Effect of Bird Strikes & Impact on Flight

In [34]:
```
plt.figure(figsize=(12, 6))
ax=sns.countplot(data=df, x='Effect: Indicated Damage')
ax.bar_label(ax.containers[0])
plt.xlabel('Effect: Indicated Damage')
plt.ylabel('Number of Bird Strikes')
plt.title('Effect of Bird Strikes on Flights')
plt.show()
```



📌23081 incidents caused no damage while 2477 incidents caused damage

In [35]:
```
plt.figure(figsize=(12, 6))
ax=sns.countplot(data=df, x='Effect: Impact to flight')
ax.bar_label(ax.containers[0])
plt.xlabel('Impact to flight')
plt.ylabel('Number of Bird Strikes')
plt.title('Effect of Bird Strikes on Flights')
plt.show()
```



📌 - 88 incidents where engine was shut down

📌 - 1121 incidents where there was precatioanry landing

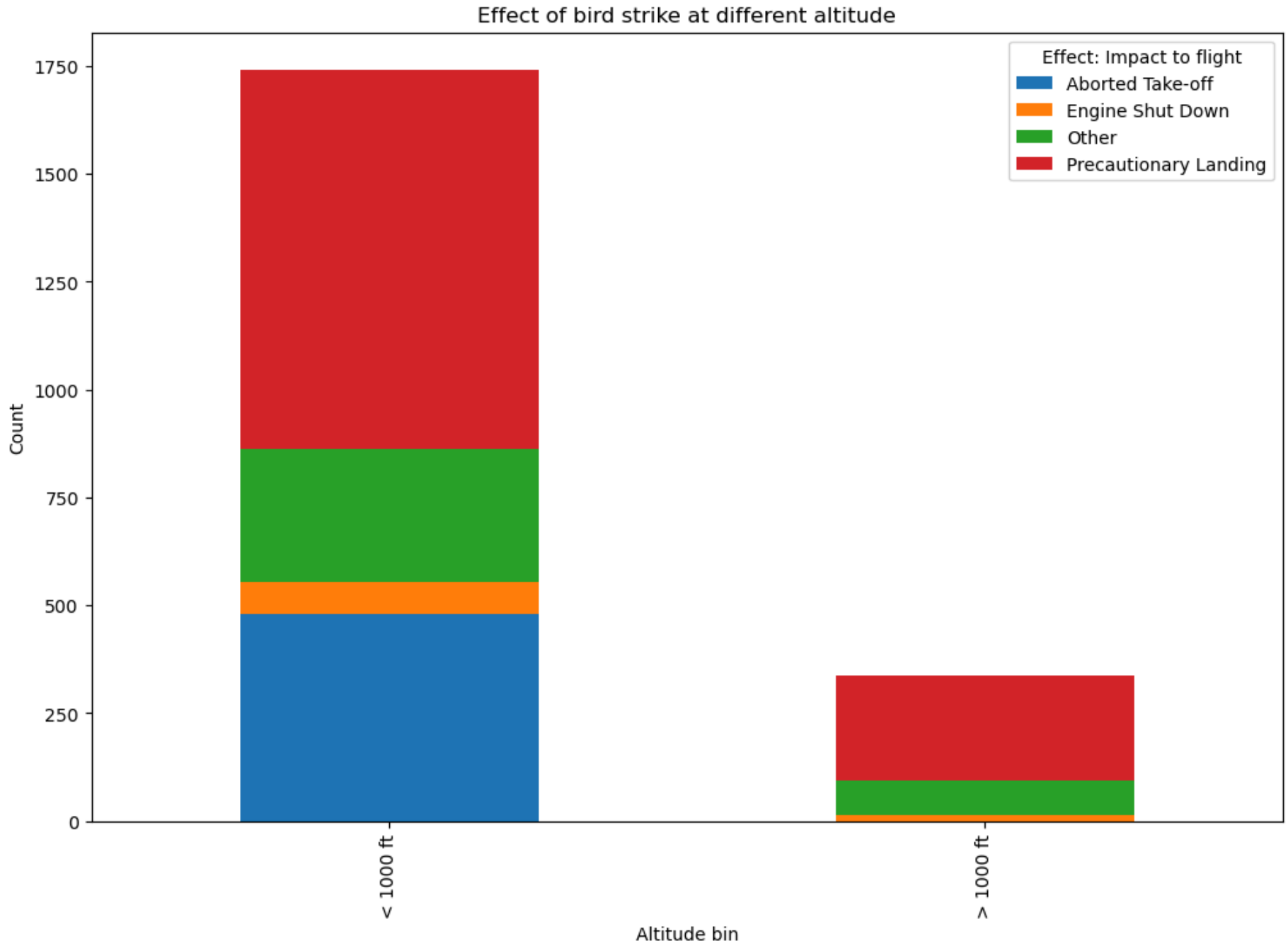
📌 - 479 incidents where take off was abortt down

- Effect of Strike at Different Altitude

In [36]:
```
altitude_effect_data=df[['Altitude bin','Effect: Impact to flight']]
```

In [37]:
```
altitude_effect_data_counts=altitude_effect_data.groupby(['Altitude bin','Effect: Impact to flight']).size().unstack()
```

In [38]:
```
altitude_effect_data_counts.plot(kind="bar",stacked=True,figsize=(12,8))
plt.xlabel("Altitude bin")
plt.ylabel("Count")
plt.title("Effect of bird strike at different altitude")
plt.legend(title='Effect: Impact to flight')
plt.show()
```


Effect of bird strike at different altitude
Effect: Impact to flight
Aborted Take-off
Engine Shut Down
Other
Precautionary Landing
Count
1750
1500
1250
1000
750
500
250
0
< 1000 ft
> 1000 ft
Altitude bin

📌The above stacked bar chart is showing the distribution of the impact of bird strikes at different altitude.

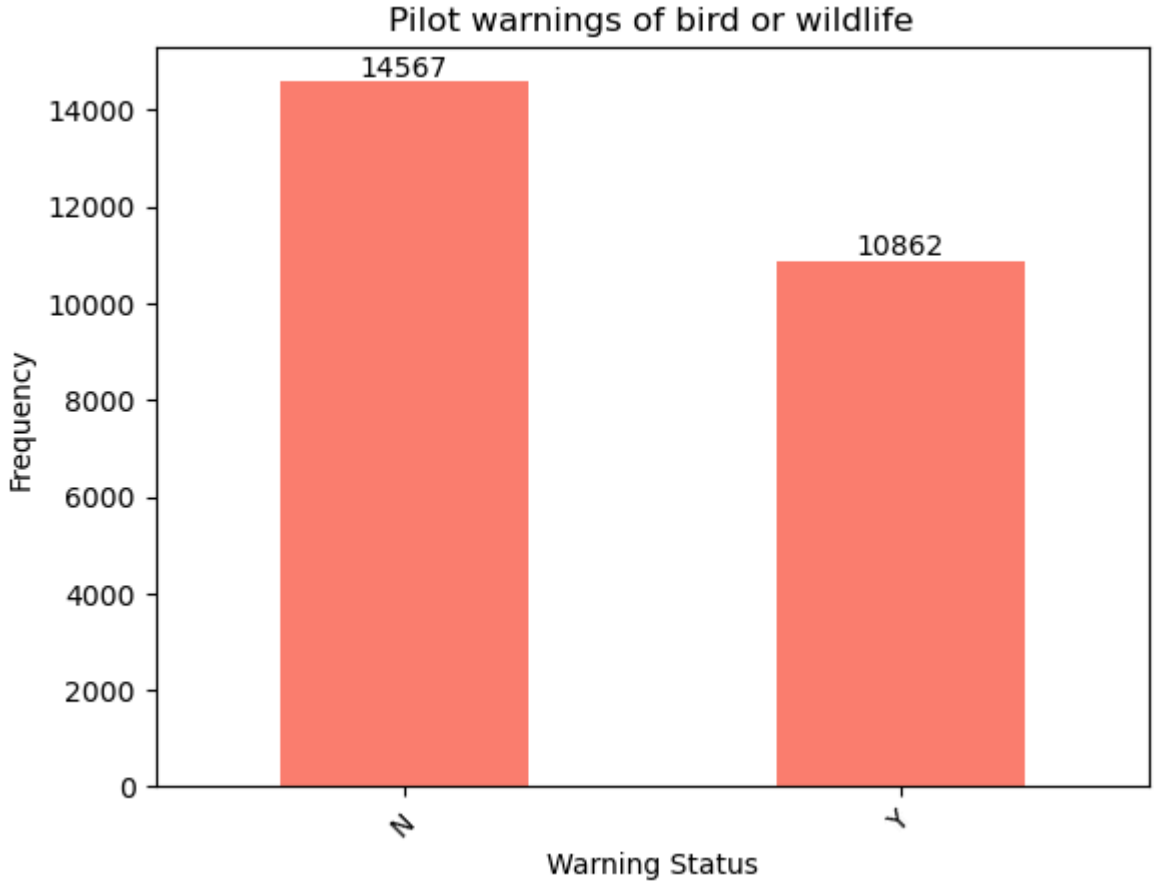
As we can see that Precautionary landing is high its

- Were Pilots Informed? & Prior Warning and Effect of Strike Relation

In [39]:
```
pilot_warnings=df['Pilot warned of birds or wildlife?']
```

In [40]:
```
warnings_count=pilot_warnings.value_counts()
```

In [41]:
```
ax=warnings_count.plot(kind='bar',color='salmon')
ax.bar_label(ax.containers[0])
plt.xlabel("Warning Status")
plt.ylabel("Frequency")
plt.xticks(rotation=45)
plt.title("Pilot warnings of bird or wildlife")
plt.show()
```

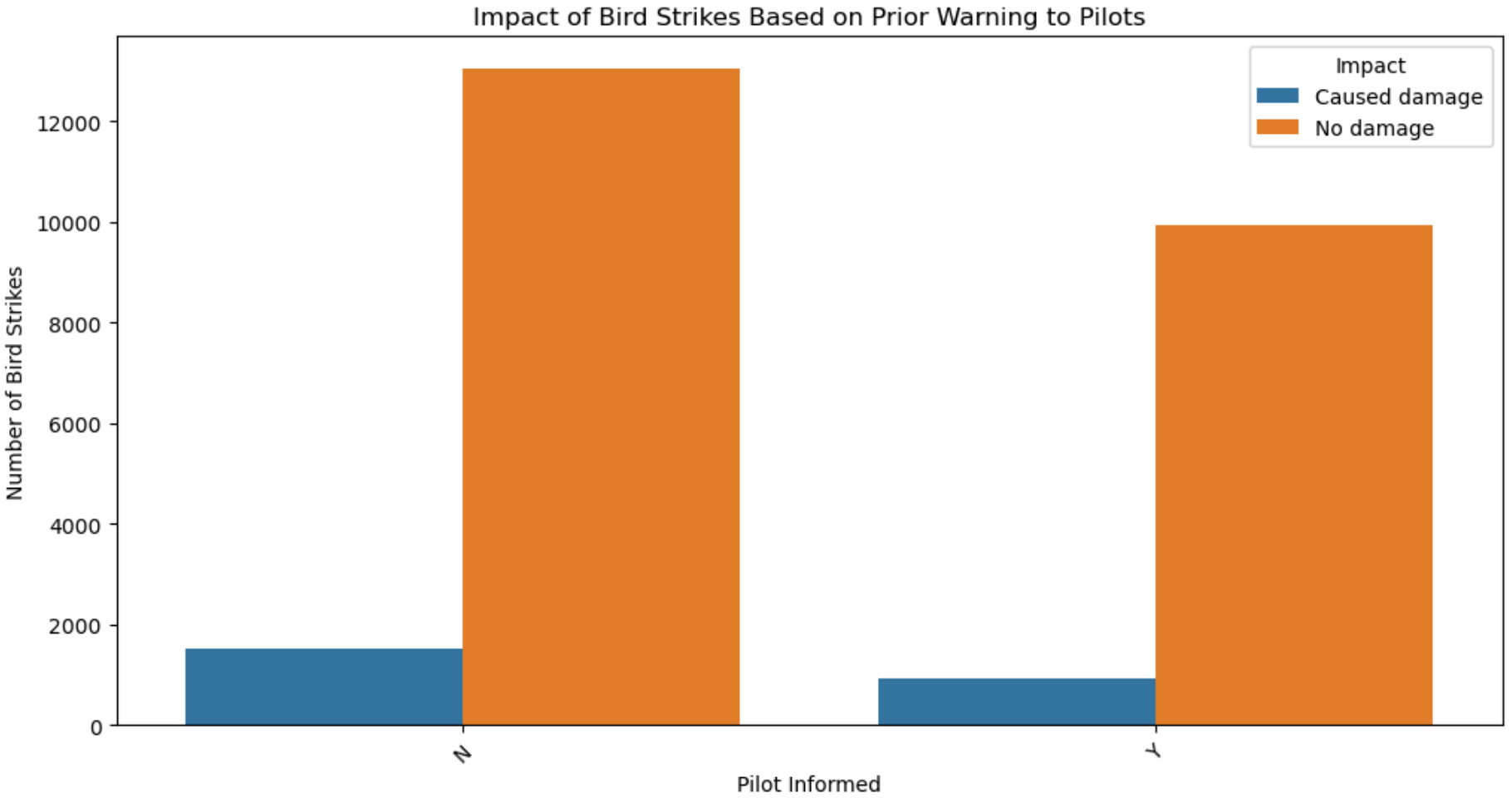


📌 10862 incidents where pilot was warned about the birds.

Relation between prior warning and effect of strike

In [42]:
```
plt.figure(figsize=(12, 6))
ax = sns.countplot(data=df, x='Pilot warned of birds or wildlife?', hue='Effect: Indicated Damage')
plt.title('Impact of Bird Strikes Based on Prior Warning to Pilots')
plt.xlabel('Pilot Informed')
plt.ylabel('Number of Bird Strikes')
plt.legend(title='Impact')
plt.xticks(rotation=45)
```

```
plt.show()
```


Impact of Bird Strikes Based on Prior Warning to Pilots
Impact
Caused damage
No damage
Number of Bird Strikes
0
2000
4000
6000
8000
10000
12000
N
Y
Pilot Informed

In [ ]: