In [1]:
```
!pip install mysql-connector-python
```

```
Requirement already satisfied: mysql-connector-python in /Users/sujag/anaconda3/lib/python3.11/site-packages
(8.2.0)
Requirement already satisfied: protobuf<=4.21.12,>=4.21.1 in /Users/sujag/anaconda3/lib/python3.11/site-packa
ges (from mysql-connector-python) (4.21.12)
```

In [2]:
```
#example of python connecting to MySQL server and databases
#
import mysql.connector
#
from mysql.connector import Error
#
try:
    connection = mysql.connector.connect(host='127.0.0.1',
                                         database='Crime_Database',
                                         user='root',
                                         password='SSuujjaa12',
                                         auth_plugin = 'mysql_native_password')
    if connection.is_connected():
        db_Info = connection.get_server_info()
        print("Connected to MySQL Server version ", db_Info)
        cursor = connection.cursor()
        cursor.execute("select database();")
        record = cursor.fetchone()
        print("Your connected to database: ", record)
#

#
except Error as e:
    print("Error while connecting to MySQL", e)
```

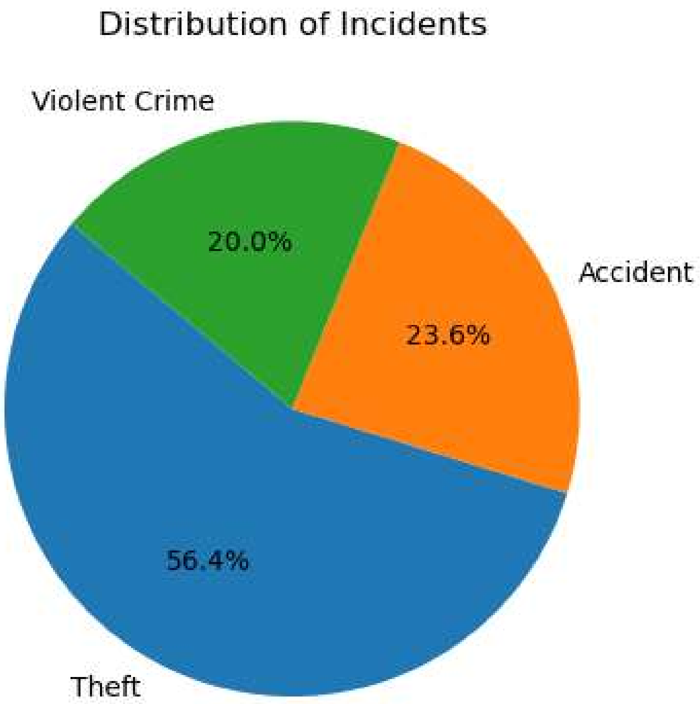
```
Connected to MySQL Server version  8.1.0
Your connected to database:  ('crime_database',)
```

In [3]:
```
#What are the types and frequencies of incidents reported?
sql_select_Query1 = '''SELECT
    INCIDENT_TYPE, COUNT(INCIDENT_ID)
FROM INCIDENTS
GROUP BY INCIDENT_TYPE;'''
cursor.execute(sql_select_Query1)
incidents_count = cursor.fetchall()
print(incidents_count)
```

```
[('Theft', 31), ('Accident', 13), ('Violent Crime', 11)]
```

In [4]:

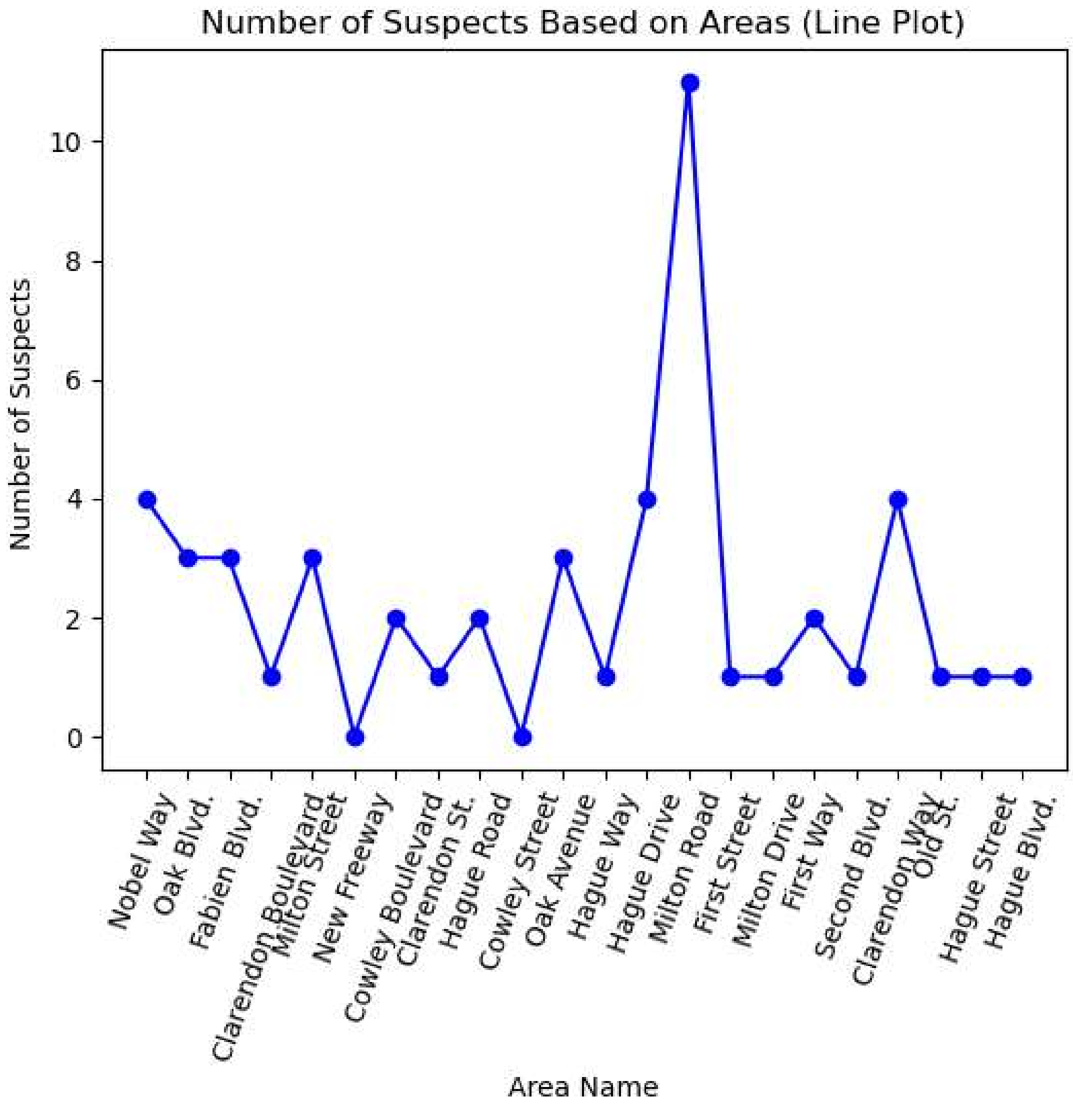
```
import matplotlib.pyplot as plt

labels, values = zip(*incidents_count)

plt.pie(values, labels=labels, autopct='%1.1f%%', startangle=140)
plt.title("Distribution of Incidents")

plt.show()
```



In [5]:

```
#How many suspects are there, and what is the distribution of their locations?
sql_select_Query2 = '''SELECT A.LOCATION, COUNT(C.SUSPECT_ID) AS SuspectCount
FROM INCIDENTS AS I
JOIN AREAS AS A ON A.Area_ID = I.Area_ID
LEFT JOIN CAUSED_BY AS C ON I.INCIDENT_ID = C.INCIDENT_ID
GROUP BY A.LOCATION;'''
cursor.execute(sql_select_Query2)
suspect_location = cursor.fetchall()
```

```python
# Extracting data for plotting
areas, incident_counts = zip(*suspect_location)

# Plotting the line chart
plt.plot(areas, incident_counts, marker='o', linestyle='-', color='b')
plt.xlabel('Area Name')
plt.ylabel('Number of Suspects ')
plt.title('Number of Suspects Based on Areas (Line Plot)')
plt.xticks(rotation=70)
plt.show()
```



In [6]:

```python
import seaborn as sns
import pandas as pd
```

In [8]:

```python
#distribution of users based on gender and ethnicity
sql_select_Query3 = '''select ethnicity, gender, count(user_id) as total_users from users
group by ethnicity, gender
order by ethnicity, gender;'''
```

```
cursor.execute(sql_select_Query3)
user_data = cursor.fetchall()
```

In [9]:

```
df=pd.DataFrame(user_data, columns=['ethnicity', 'gender', 'total_users'])
df
```

Out[9]:

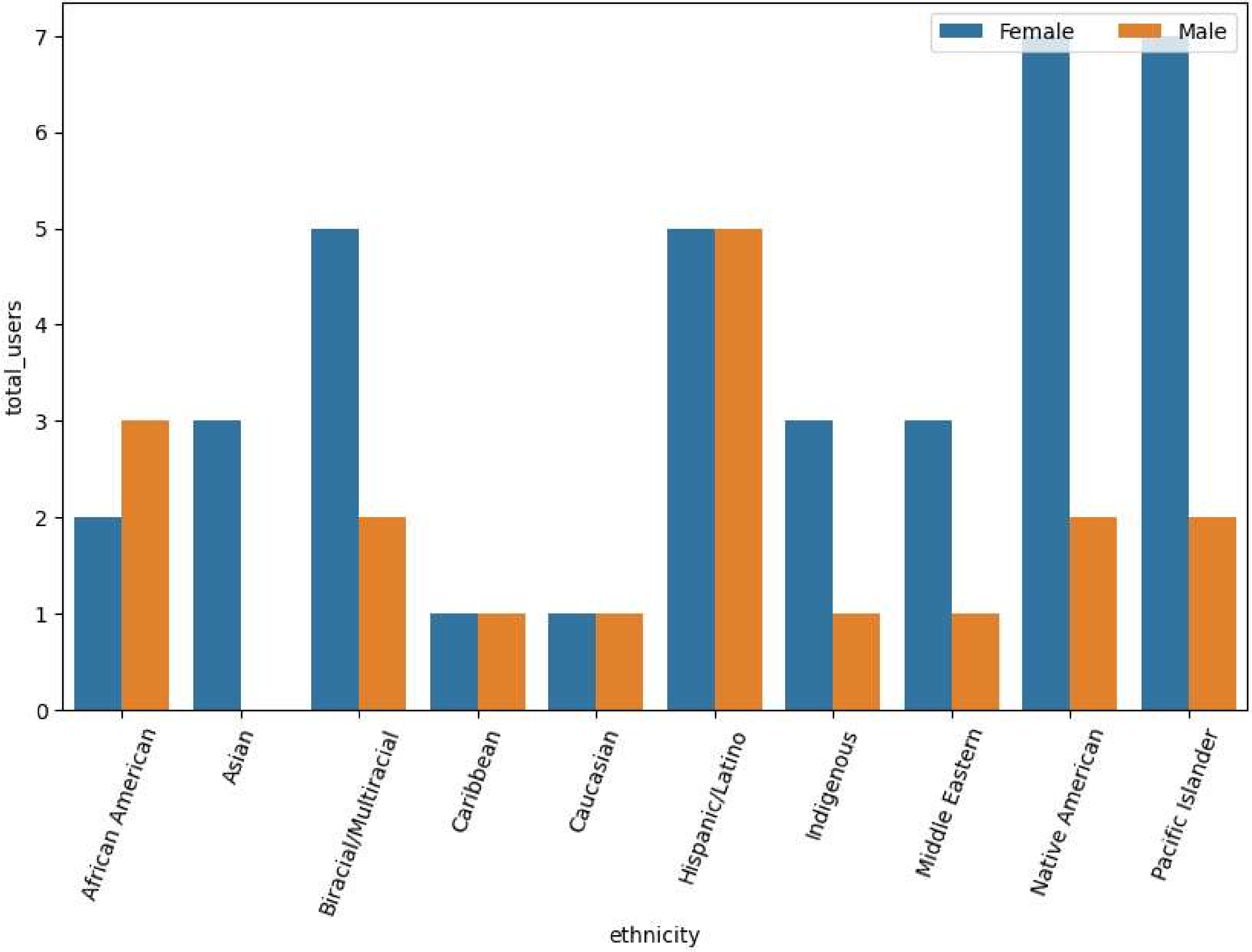
	ethnicity	gender	total_users
0	African American	Female	2
1	African American	Male	3
2	Asian	Female	3
3	Biracial/Multiracial	Female	5
4	Biracial/Multiracial	Male	2
5	Caribbean	Female	1
6	Caribbean	Male	1
7	Caucasian	Female	1
8	Caucasian	Male	1
9	Hispanic/Latino	Female	5
10	Hispanic/Latino	Male	5
11	Indigenous	Female	3
12	Indigenous	Male	1
13	Middle Eastern	Female	3
14	Middle Eastern	Male	1
15	Native American	Female	7
16	Native American	Male	2
17	Pacific Islander	Female	7
18	Pacific Islander	Male	2

In [10]:

```
plt.figure(figsize=(10, 6))
ax = sns.barplot(x = 'ethnicity',
                  y = 'total_users',
                  hue='gender',
                 data=df)
ax.legend(loc = 'upper right',
          ncol = 4)
plt.xticks(rotation=70)
```

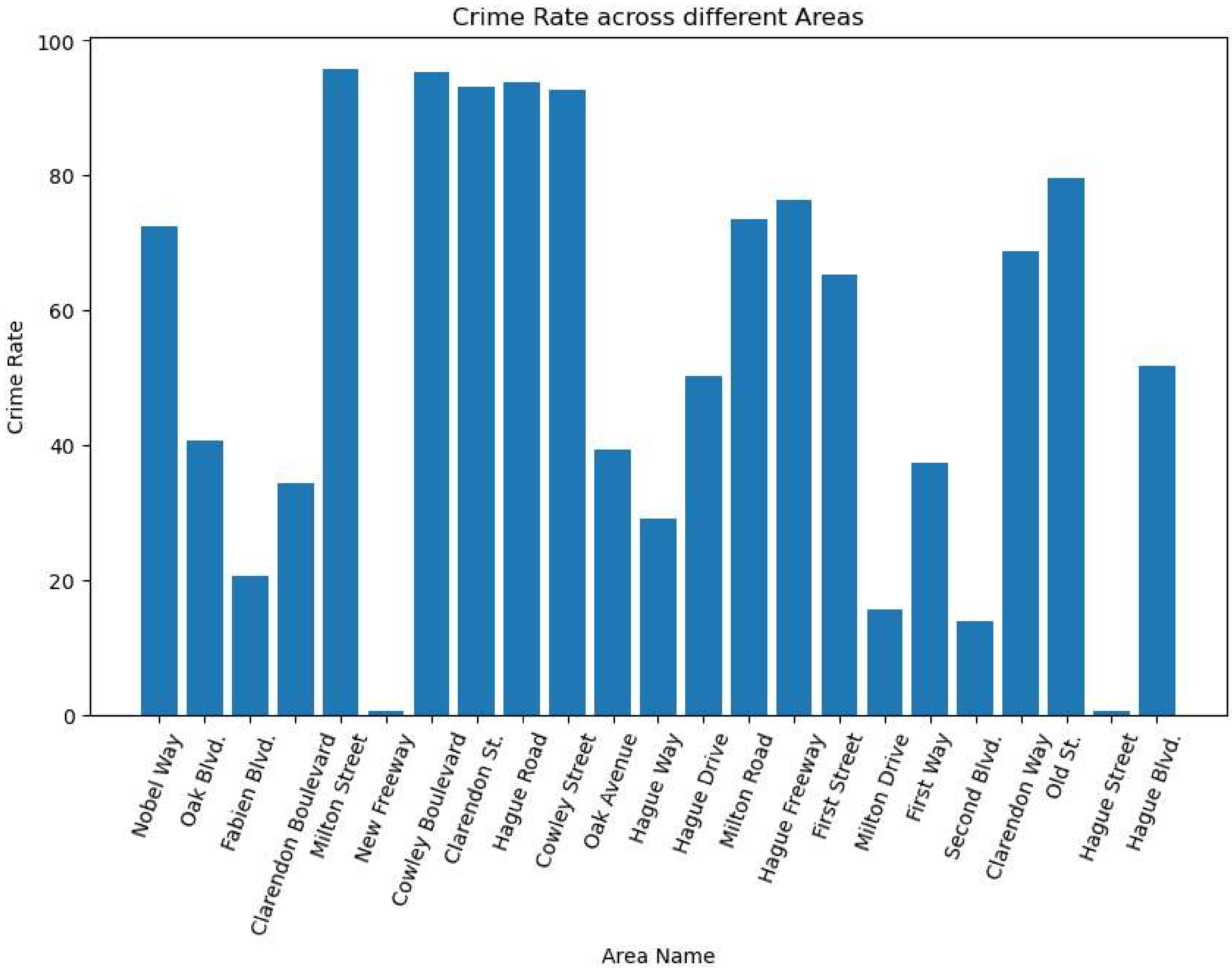
Out[10]:

```
(array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
 [Text(0, 0, 'African American'),
  Text(1, 0, 'Asian'),
  Text(2, 0, 'Biracial/Multiracial'),
  Text(3, 0, 'Caribbean'),
  Text(4, 0, 'Caucasian'),
  Text(5, 0, 'Hispanic/Latino'),
  Text(6, 0, 'Indigenous'),
  Text(7, 0, 'Middle Eastern'),
  Text(8, 0, 'Native American'),
  Text(9, 0, 'Pacific Islander')])
```



```
In [11]: # crime rate across different areas
         plt.figure(figsize=(10, 6))
         sql_select_Query4 = '''SELECT Location, crime_rate
         FROM areas;'''
         cursor.execute(sql_select_Query4)
```

```
data4 = cursor.fetchall()
Location, Crime_Rate = zip(*data4)

# Plotting the line chart
plt.bar(Location, Crime_Rate)
plt.xlabel('Area Name')
plt.ylabel('Crime Rate')
plt.title('Crime Rate across different Areas')
plt.xticks(rotation=70)
plt.show()
```



In [12]:

```
# Correlation between crime rates and the occurrence of incidents in specific areas
plt.figure(figsize=(10, 6))
sql_select_Query5 = '''select a.Crime_Rate, count(i.incident_id) as total_incidents
```

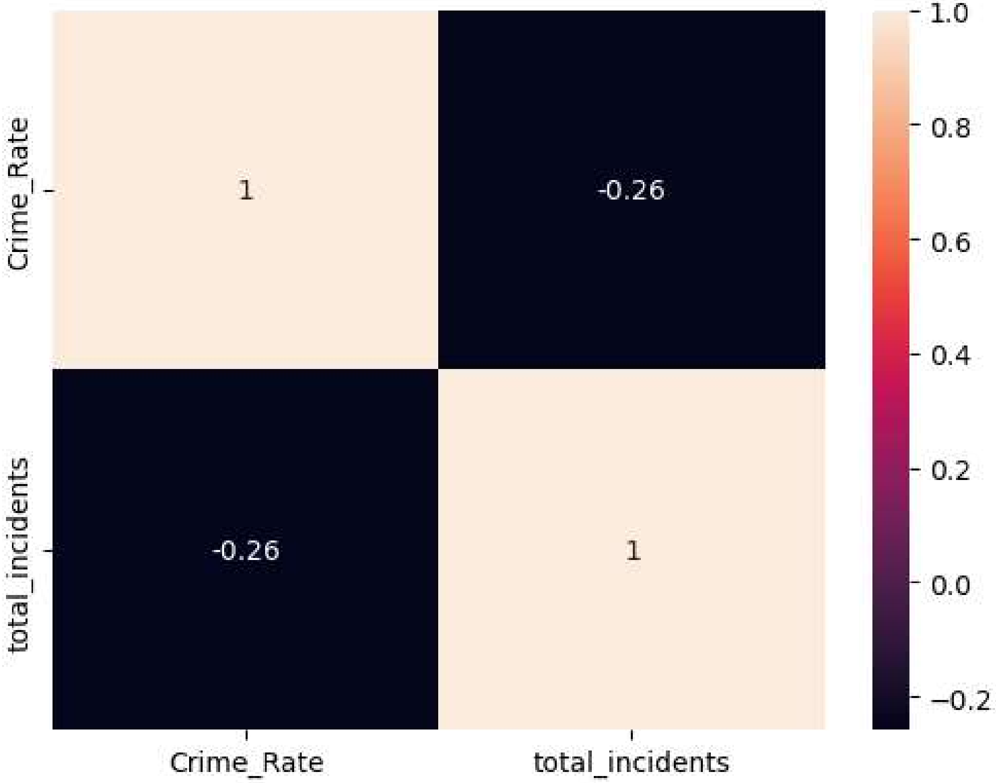
```
from areas a
inner join incidents i on a.area_id=i.area_id
group by a.area_id;'''
cursor.execute(sql_select_Query5)
data5 = cursor.fetchall()

df2=pd.DataFrame(data5, columns=['Crime_Rate', 'total_incidents'])
df2
```

Out[12]:

	Crime_Rate	total_incidents
0	72.36	1
1	40.60	5
2	20.56	3
3	32.94	4
4	95.81	1
5	0.61	1
6	95.41	1
7	93.05	1
8	93.77	2
9	92.77	1
10	39.32	3
11	34.40	3
12	29.11	1
13	50.14	5
14	11.48	5
15	65.33	1
16	15.52	1
17	37.44	2
18	13.84	1
19	68.80	4
20	79.72	2
21	0.50	2
22	51.79	3
23	73.65	2

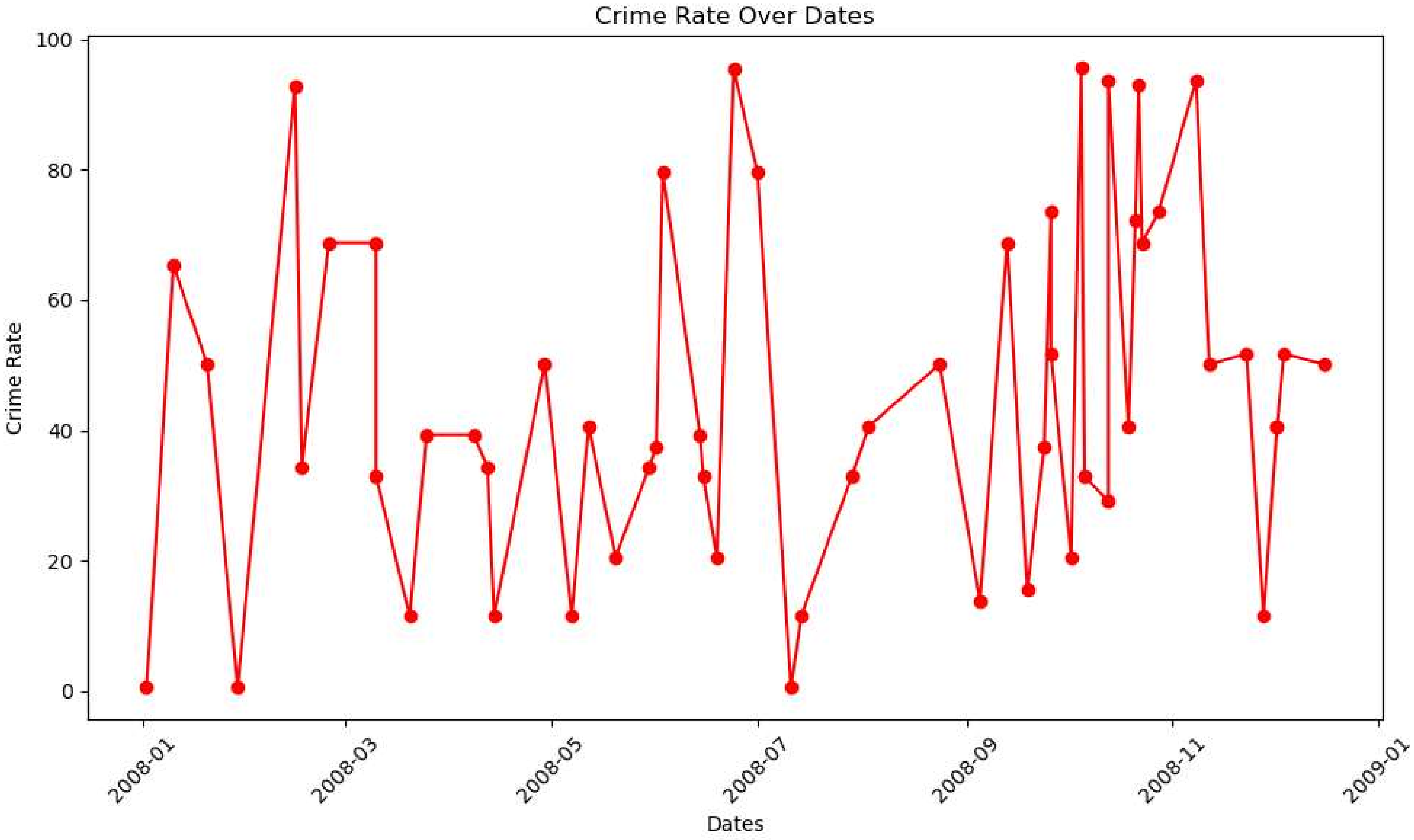
```
<Figure size 1000x600 with 0 Axes>
```

In [13]:
```
#Correlation between crime rates and the occurrence of incidents in specific areas
corr_matrix = df2[['Crime_Rate', 'total_incidents']].corr()
sns.heatmap(corr_matrix, annot=True)
plt.show()
```



In [14]:
```
from datetime import datetime
import pandas as pd
#overall trend in crime rates over the observed time period
sql_select_Query5 = '''SELECT
    a.crime_rate, DATE(i.Incident_DateTime)
FROM
    incidents i,
    areas a
WHERE
    a.Area_ID = i.Area_ID;'''
cursor.execute(sql_select_Query5)
crimerate_per_time = cursor.fetchall()
#print(crimerate_per_time)
```

```
# Process the result
dates = []
crime_rates = []

for row in crimerate_per_time:
    crime_rate, incident_date = row
    dates.append(datetime.strftime(incident_date, '%Y-%m-%d'))
    crime_rates.append(crime_rate)

crimerates_per_data =pd.DataFrame({
    'Dates': dates,
    'Crime_Rate': crime_rates
})

crimerates_per_data['Dates'] = pd.to_datetime(crimerates_per_data['Dates'])  # Convert Timestamp to datetime

# Sort DataFrame by Timestamp
crimerates_per_data.sort_values(by='Dates', inplace=True)

# Plotting the line graph
plt.figure(figsize=(10, 6))
plt.plot(crimerates_per_data['Dates'], crimerates_per_data['Crime_Rate'], marker='o', linestyle='-', color='r
plt.xlabel('Dates')
plt.ylabel('Crime Rate')
plt.title('Crime Rate Over Dates')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```


Crime Rate Over Dates
Crime Rate
Dates
0
20
40
60
80
100
2008-01
2008-03
2008-05
2008-07
2008-09
2008-11
2009-01